1-1-2016

A Comparison Between Propensity Score Matching, Weighting, and Stratification in Multiple Treatment Groups: A Simulation Study

Priyalatha Govindasamy
University of Denver

A Comparison Between Propensity Score Matching, Weighting, and Stratification in Multiple Treatment Groups: A Simulation Study

Abstract

The application of propensity score techniques (matching, stratification, and weighting) with multiple treatment levels are similar to those used in binary groups. However, given that the application of propensity scores in multiple treatment groups is new, factors affecting the performance of matching, stratification, and weighting in multiple treatment groups are less explored. Therefore, this study was conducted to determine the performance of different propensity score techniques with multiple treatment groups under various circumstances. Specifically, the study focused on examining how the three propensity score corrective techniques perform in estimating treatment effects under (1) overt and (2) hidden types of selection bias. In this study, the performance of propensity score matching, stratification, and weighting techniques were tested under three different sample sizes and three levels of overt and hidden bias.

A Monte Carlo simulation was used to generate data with specific sample sizes and levels of overt and hidden bias. A total of 54 data conditions with 1000 replications for each condition was generated to compute the average treatment effect (ATE). The difference between the pre-specified ATE and estimated ATE was calculated to evaluate the performance of propensity score techniques. Two 3x3x3x2 analyses of variance were conducted to assess the effect of propensity score technique, level of bias, sample size, and type of treatment effect on the amount of bias in estimating the treatment effect under overt and hidden bias conditions.

The results provided four key findings of information about the application of propensity score analysis in multiple treatment groups. The first key finding is that the treatment effect estimate will be underestimated after imposing propensity score adjustments. Second, the treatment effect estimates are affected by the level of overt bias. Third, propensity score analysis does not account for hidden bias. The fourth finding is that the propensity score techniques performed differently in a small sample size condition. Overall, these four key findings provide cautionary notes to the users of propensity score analysis in multiple treatment groups. The study is concluded with the limitations of this study and the recommendations for future research.

Keywords: Propensity score, multiple treatment

Document Type

Dissertation

Degree Name

Ph.D.

Department

Quantitative Research Methods

First Advisor

Antonio Olmos, Ph.D.

Second Advisor

Kathy Green

Third Advisor
Kellie Keeling

Keywords
Multiple treatment, Propensity score, Monte Carlo simulation

Subject Categories
Educational Assessment, Evaluation, and Research | Statistics and Probability

A COMPARISON BETWEEN PROPENSITY SCORE MATCHING, WEIGHTING, AND STRATIFICATION IN MULTIPLE TREATMENT GROUPS: A SIMULATION STUDY

A Dissertation

Presented to

the Faculty of the Morgridge College of Education

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

Priyalatha Govindasamy

August 2016

Advisor: Antonio Olmos, Ph.D

Author: Priyalatha Govindasamy
Title: A COMPARISON BETWEEN PROPENSITY SCORE MATCHING, WEIGHTING, AND STRATIFICATION IN MULTIPLE TREATMENT GROUPS: A SIMULATION STUDY
Advisor: Antonio Olmos, Ph.D
Degree Date: August 2016

Abstract

The application of propensity score techniques (matching, stratification, and weighting) with multiple treatment levels are similar to those used in binary groups. However, given that the application of propensity scores in multiple treatment groups is new, factors affecting the performance of matching, stratification, and weighting in multiple treatment groups are less explored. Therefore, this study was conducted to determine the performance of different propensity score techniques with multiple treatment groups under various circumstances. Specifically, the study focused on examining how the three propensity score corrective techniques perform in estimating treatment effects under (1) overt and (2) hidden types of selection bias. In this study, the performance of propensity score matching, stratification, and weighting techniques were tested under three different sample sizes and three levels of overt and hidden bias.

A Monte Carlo simulation was used to generate data with specific sample sizes and levels of overt and hidden bias. A total of 54 data conditions with 1000 replications for each condition was generated to compute the average treatment effect (ATE). The difference between the pre-specified ATE and estimated ATE was calculated to evaluate the performance of propensity score techniques. Two 3x3x3x2 analyses of variance were conducted to assess the effect of propensity score technique, level of bias, sample size, and type of treatment effect on the amount of bias in estimating the treatment effect under overt and hidden bias conditions.

The results provided four key findings of information about the application of propensity score analysis in multiple treatment groups. The first key finding is that the treatment effect estimate will be underestimated after imposing propensity score adjustments. Second, the treatment effect estimates are affected by the level of overt bias. Third, propensity score analysis does not account for hidden bias. The fourth finding is that the propensity score techniques performed differently in a small sample size condition. Overall, these four key findings provide cautionary notes to the users of propensity score analysis in multiple treatment groups. The study is concluded with the limitations of this study and the recommendations for future research.

Keywords: Propensity score, multiple treatment

Acknowledgement

This endeavor would not be possible without the help and support of many kind people around me. First and foremost, I would like to thank God (my Vinayagar) for his continuous blessings throughout my journey. My sincere thanks to the Ministry of Higher Education and Sultan Idris Education University, Malaysia for sponsoring my studies. The completion of this dissertation could not have been possible without my mentors, Dr Antonio Olmos, Dr Kathy Green and Dr Kellie Keeling. Throughout the program, I was very grateful for having excellent professors who guided and trained me to become a researcher and teacher. Dr. Olmos's advice, patience, unsuppressed knowledge, guidance and care were incredibly supportive in completing this dissertation. Dr. Green's patience, care and her hug always had a motherly touch in addition to being an academic advisor. , I also need to acknowledge my greatest advocates, mum (Mrs. Tanapakiam Chelliah) and dad (Mr Govindasamy Nadarajo) for their unconditional love, blessing, support and tolerance to my statistical jargon. My heartfelt thanks to my sisters (Moganes, Manonmani), brother-in laws (Badmanathan, Karnakaran) and all my nieces and nephews (Tharshini, Yuven, Vaishu, Yoogen & Thaneesha) for being my cheerleading team. I also want to thank my colleagues in the RMS program, Maria Vukovich, Tara Rhodes, Holly Roof, Ksenia Polson, Elizabeth Anderson and Melissa Hoffman. My special thanks to Maria for all your help with my writing and continuous moral support. Thanks to Ms Jackie Carrillo, Dr Lin Ma, Ms Gabriel Hottinger, Dr Mariana Enriquez and Dr Cindy McRae for your moral support, advice and well wishes that always empowered my self-confidence. Finally, I dedicate this achievement to my late grandfather, Mr Chelliah.

Table of Contents

List of Tables

List of Figures

Chapter One: Background and Literature Review

Introduction

Chapter one provides the background for the study and a review of the relevant literature. I begin with a brief introduction to the study. The introduction presents the relationship between causality and propensity scores. Next, the rationale for this study is presented. In this chapter, I explicitly state my objectives and the research questions that I address through this study. I state the significance of this study and the limitations. Next, I define the terms that I use regularly in this study.

I begin the literature review with a description of selection bias. Subsequently, multivariable approach and propensity score analysis techniques in controlling for selection biases are discussed. In this study, I focus on the application of propensity score analysis in controlling for selection biases. I explain the three common applications of propensity score techniques which are (1) propensity score matching, (2) stratification using propensity scores, and (3) propensity score weighting. Then, the application of propensity scores in multiple treatment groups is reviewed, followed by a review of the different directions of propensity score applications in multiple treatment groups. The application of propensity scores in multiple treatment assignments is presented. A summary of propensity scores in multiple treatment groups concludes this chapter.

Background

Causal inference means making causal claims of an unknown population from a known sample (Morgan & Winship, 1999). Empirical evidence from a causal claim helps to generalize the effect as a result of cause to a larger population (Gelman, 2011). According to Lazarfeld (1959), three criteria must be met to establish causality. The three criteria are (1) cause precedes the effect (temporal precedence), (2) cause is related to the effect (statistical relationship between the variables) and, (3) ruling out potential explanations to causation (spurious relationship). When these three conditions are met, causality can be estimated as the difference between the outcomes of individuals who received treatment (treated) versus the potential outcome for the same individuals had they not received treatment (not treated). In other words, treatment effect is the difference in the outcome for an individual/observation assigned to treated and non-treated groups at the same time in the same context. The potential outcome that would have been observed under different exposure for the same individual is called a counterfactual (Guo & Fraser, 2015). According to Neyman-Rubin's counterfactual framework, a person's observed outcome is a combination of two outcomes. It can be mathematically stated as follows:

$$Y_i = W_i Y_{1i} + (1 - W_i) Y_{0i} \qquad (1)$$

In equation 1, W_i = is treatment assignment which can be 1 or 0, Y_1 represents the outcome in the treatment group while Y_0 represents the outcome in the control group.

When an individual is assigned to a group, only one of the outcomes is observed. This is referred to as a "fundamental problem of causal inference" (Guo & Fraser, 2015).

The idea is similar to the concept of missingness. The unobserved outcome for an individual is assumed to be a missing value and researchers try to estimate it with the observed information.

In randomized experiments, individuals are randomly assigned to treatment conditions. Randomization provides an equal chance of being assigned to treated and non-treated groups. Randomization ensures that the treated and non-treated groups are probabilistically similar prior to any treatment. This makes the observations in the non-treated group the counterfactuals for the treated and vice versa. But randomization is not always possible or ethical. Therefore, quasi-experiments are proposed and used as alternatives to randomized designs. Quasi-experiments as do not employ randomization in selecting the sample for each group. This makes the covariates for the treated and non-treated groups probabilistically unequal. Due to the unequal distribution of the samples in treated and non-treated groups, exploring causation using quasi-experiments provides "less compelling support for counterfactual inferences" (Shadish, Cook & Campbell, 2002, p.14). This is because the estimation of treatment effect in non-randomized designs becomes biased and inefficient (Shadish et al., 2002).

This draws attention to the need to have good counterfactuals between the treated and non-treated groups in quasi-experimental designs. In other words, researchers need to make adjustments to the comparison groups in quasi-experimental designs so that they are homogenous in terms of the distribution of the characteristics. If the groups are homogenous, then the members or observations within each group are assumed to be similar. The homogenous groups also assure us that the observations in the control group are useful as counterfactuals to those in treatment and *vice versa*. To do that, researchers find observations in control groups that are similar in terms of the observed characteristics to observations in treatment. The process of finding observations is really finding counterfactuals for observations in the treatment group. Researchers use statistical approaches to obtain groups that are homogenous with compelling counterfactual quality.

Statistical approaches such as ordinary regression, covariate adjustment analysis, structural equation modeling, selection models, and matching methods can be applied to adjust for differences between the groups in non-randomized designs (Shadish et al., 2002; Stuart, 2010). The statistical adjustments to the comparison groups help to control for selection biases. Propensity score analysis is one of the recent developments under applied statistical methods addressing causal effects in non-random designs. It has been found that the use of propensity scores is able to reduce selection bias and increase precision in causal estimation (D'Agostino, 1998). A propensity score is the conditional probability of treatment given the observed covariates (Austin, 2011). Conditional on measured baseline covariates, allocation of individuals to treatment groups is considered to be a random process that mimics randomized designs (Austin, 2011; Sturmer, 2006). This is because observations in treated and non-treated groups with equal or similar propensity scores have a nearly similar background distribution of the covariates (D'Agostino, 1998). When the groups are comparable, the distributions of the observed baseline covariates are expected to be similar across the groups. Once propensity scores are estimated, they can be used in various ways. Typically, propensity scores are implemented in matching, stratification, and weighting techniques (Olmos & Govindasamy, 2015).

Researchers are as of 2015 actively discussing and investigating applications of the different propensity score techniques. The merits of implementing different propensity score techniques are the subject of an on-going debate among researchers (An, 2010). Researchers are currently exploring the performance of different propensity score techniques to identify best practices in propensity score applications. Best practices are intended to provide guidance for practitioners in implementing propensity score analysis.

Problem Statement

Propensity score analysis is a multi-step procedure used to equate groups for comparisons whose purpose is to reduce bias associated with non-randomization (Lane, 2011). Generally propensity score analysis is performed through matching, stratification, and weighting (Caliendo & Kopeinig, 2008). Propensity scores are estimated the same way for all three techniques, but the way techniques are implemented to address selection bias differs (D’Agostino, 1998). In matching, a propensity score is used to find matches between the treated and non-treated cases. Propensity scores are stratified to categorize treated and non-treated observations into strata with the same propensity score range. A propensity score can also be used as a weight to account for non-constant variability on the observed covariates between treated and non-treated groups. Given the differences in the implementation of propensity scores, researchers are unclear on the merits of selecting an appropriate propensity score technique (Luellen, 2007).

Researchers have compared the performance of the propensity score techniques under different data conditions for binary groups (Mitra & Reiter, 2012; Wilde & Hollister, 2007). The findings from the comparison studies have been informative for practitioners in selecting appropriate propensity score analysis techniques from the pool of propensity score techniques for binary groups.

Until recently, propensity score analysis was studied for binary (treated versus non-treated) groups. Imbens (2000) and Lu, Greevey, Xu, and Beck (2011) have explained the application of propensity scores in multiple treatment groups. These researchers introduced the same propensity score techniques (matching, stratification, and weighting) used in binary groups to groups with multiple treatment levels. Since the application of propensity scores in multiple treatment groups is new, the performance of matching, stratification, and weighting in multiple treatment groups is less explored. Also, the literature on propensity score application in multiple treatment groups is limited. The limited literature on the application of propensity scores to multiple treatment groups motivates an exploration of the application and performance of propensity scores with multiple treatment groups.

Purpose of the Study

The purpose of the study was to compare the performance of matching, stratification, and weighting techniques using propensity score analysis in multiple treatment exposure groups with simulated data. The goal was to demonstrate how three propensity score corrective techniques for multiple treatment groups perform in estimating treatment effects under different types of selection biases. Monte Carlo simulation was used to generate data. The performance of the three approaches (matching, stratification, and weighting) correcting (1) overt and (2) hidden selection bias conditions was assessed using the amount of bias introduced in the average treatment effect as the outcome measure. Again, the amount of bias in the average treatment effect estimate was used to determine the influence of sample size in propensity score corrective approaches accounting for overt and hidden biases. Therefore, the specific research questions of this study were as follows:

1. Which of the three techniques (matching, weighting, and stratification) performs the best in the presence of overt selection bias?

2. Which of the three techniques (matching, weighting, and stratification) performs the best in the presence of hidden selection bias?

3. Does sample size (small, medium, and large) influence the performance of matching, stratification, and weighting techniques under overt and hidden selection biases?

4. Do varying degrees of overt and hidden biases influence the performance of matching, stratification, and weighting techniques under different sample sizes?

The ultimate goal of propensity score analysis is to obtain comparable groups which will lead to unbiased treatment effect estimates. Therefore, the degree of bias in treatment effect was computed to determine the best of three propensity score corrective approaches accounting for overt and hidden biases under small, medium, and large sample sizes. First, the overt and hidden bias conditions are defined. Second, I describe the requirements and steps in setting overt and hidden bias conditions for the study. Next, I specify steps taken in generating three different sample sizes for the study. Monte Carlo simulation was then used to generate data with the conditions specified in step 2. Then, I discuss steps in performing all three propensity score corrective approaches (matching, stratification, and weighting). The average treatment effect estimate (ATE) were computed after correcting for biases. Then, the difference between the true and estimated average treatment effect (ATE) was computed. Finally, a four-way analysis of variance (ANOVA) was used to assess the performance of propensity score techniques, sample size, type of treatment effect, and the levels of overt and hidden bias on the difference between true and estimated ATE. The ANOVA was used to capture the performance of propensity score techniques under various conditions. Also, the interaction between the propensity score technique and level of overt or hidden bias was used to compare the performance of all three corrective approaches under varying degrees of selection bias. In the same way, the interaction between propensity score techniques and sample size from ANOVA was assessed to understand the influence of sample size on the performance of propensity score techniques.

Significance of the Study

Propensity score analysis in multiple treatment groups is a fairly new approach and has received less attention than analysis with two groups. By examining the performance of propensity score analysis in multiple treatment groups, this study is expected to be a contribution for practitioners applying propensity score analysis. Comparing propensity score techniques in multiple treatment groups distinguishes this work from previous studies that focused on comparing the techniques in binary groups. In addition, the investigation of propensity score analysis under overt and hidden bias in multiple treatment groups makes this study unique.

Limitations

In this simulation study, some conditions are pre-defined and held constant. In generating the data, all the variables were set to be continuous and normally distributed. Next, the correlation between the generated variables was set to be less than or close to .20) to represent negligible relationship between them. The settings of the R packages (TriMatch and twang) that were used for propensity score analysis is another limitation to this study. Table 1 summarizes the pre-defined settings in the application of propensity score techniques in multiple treatment groups.

Table 1

Default Setting in the Application of Propensity Score Techniques

Propensity score technique	**Package**	**Setting**
Propensity score matching	TriMatch	• Maximumtreat: The package creates matches with replacement as a default • Caliper: Caliper size is 0.25 of standard deviation of the propensity score as a default
Stratification using propensity score	None	• Five equally distributed strata will be used in applying stratification
Propensity score weighting	Twang	• Twang by default estimates propensity scores using Generalized Boosted Model (GBM). • Number of trees for the classification purpose in GBM will be set at 3000 as the default.

Definitions of Terms

Monte Carlo simulation. Monte Carlo (MC) computer simulation is a process of repeatedly generating random samples and performing statistical analysis to estimate results (Raychuadhuri, 2008). MC methods are set up as an experiment, where data are generated to test specific theoretically derived hypotheses (Paxton, Curran, Bollen, Kirby, & Chen, 2001).

Propensity score analysis. Propensity score analysis is a multi-step procedure used to equate groups of comparisons that reduces bias due to non-randomization (Lane, 2011). The steps can be illustrated as covariate selection and propensity score estimation, balance check, utilization of propensity score model (e.g., matching, stratification, or weighting), balance check after accounting for selection bias, and finally estimation of the treatment effect similar to random designs (Austin, 2008, D'Agostino, 1998; Luellen, Shadish & Clark, 2005).

Overt bias. Overt bias is a type of selection bias. Selection bias introduces differences between the comparison groups. If the source of differences can be identified and measured, then it is called overt bias. Overt bias can be observed in the data prior to any treatment implementation (Rosenbaum, 2002). Since the source bias can be identified in the data, statistical adjustments can be performed to control the bias.

Hidden bias. Hidden bias is an unknown source of bias that cannot be measured. Selection bias introduces differences between the treatment and comparison groups. If the source of differences cannot be identified and measured, then it is called hidden bias, and the reason for the groups to differ cannot be determined. Not explicitly knowing the source of bias makes it harder to control for it in the data.

Multiple treatment variable. A multiple treatment variable is a nominal variable with more than two levels of treatment assignment. In this study, the multiple treatment variable consisted of three levels and they were (1) treatment 1, (2) treatment 2, and (3) control. Treatment 1 differed from treatment 2 by the number of observations. The number of observations in treatment group 1 was set higher than in treatment group 2 to mimic a real data context.

Literature Review

Selection Bias

Selection bias is one of the major methodological challenges in observational research (Starks, Diehr, & Curtis, 2009). Selection bias is the influence of non-randomization in the enrollment of subjects into treatment and control groups. This becomes a problem in an observational study because samples in the treatment and control group will be different in their characteristics. When the characteristics of the sample that predispose selection into treatment are related to the outcome, it introduces confounding from selection bias. Confounding variables includes both measured and unmeasured factors related to both treatment and the outcome. The consequences of ignoring confounding in the analysis can result in inflated Type 1 error rates, where the effects of treatment are falsely attributed to the intervention (Starks et al., 2009). Thus, the treatment effect estimate will differ from its true value and be biased. Negative or null bias illustrates closeness to the true value whereas positive bias means the estimated parameter is greater than the true value (Rodriguez & Llorca, 2004). The direction of bias from the estimated parameter indicates presence or absence of bias in the estimation.

Selection bias can be categorized as (1) overt and (2) hidden bias (Rosenbaum, 2002). Overt bias means that the source of bias in the data can be identified and measured (Braitman & Rosenbaum, 2002). When the source of bias is unknown, then it is referred to as hidden bias (Braitman & Rosenbaum, 2002). The overt type of selection bias in an observational study can be addressed using statistical adjustments (Rosenbaum, 2002). Correcting for overt selection bias is only possible when the variables influencing selection are measured in a study (Rodriguez & Llorca, 2004). Literature suggests controlling the variables influencing selection using two approaches: (1) multivariable regression modelling, and (2) propensity score analysis (Starks et al., 2009). Following is a description of the two statistical techniques used for controlling selection bias.

Controlling selection bias using a multivariable approach. Regressing the outcome on confounding variables will control for the effects of confounding variables when estimating the contribution of the treatment. In regression, the coefficients of the treatment variables are estimated after holding other variables constant. The true impact of the treatment effect can be estimated after controlling for the effects of other variables. The choice of variables to include in the model will depend on the research question, sample size, and the availability of measures of the relevant variables. The objective is to include a set of variables that are theoretically or actually correlated with both the treatment and outcome to reduce the bias in the estimate of the treatment effect (Haro et al., 2006; Wunsch, Zwrible, & Angus, 2006).

Including all potential confounders in the regression may decrease the bias of the treatment effect. But adding more variables can decrease statistical power in small samples because it increases the error (spread) around the regression estimate by reducing the number of degrees of freedom. The goal of model building is to carefully select the best sets of confounding variables that include the most important factors likely to account for differences between intervention and comparison groups and achieve a balance in the trade-off between bias and variance in order to obtain more precise estimates of the treatment effects. It is good practice to model the entire set of confounding variables in the regression. But this practice has implications when the sample size is small. There is always a trade-off between reduced bias and increased variance around the regression estimate due to reduced degrees of freedom.

Controlling selection bias using propensity score analysis approach. Propensity score modelling is another technique for controlling confounding effects in observational designs (Starks et al., 2009). The process of controlling for confounding effects is similar to multivariable regression modelling except propensity score analysis models the treatment assignment prior to predicting the outcome. Propensity score analysis fits a model to predict the treatment assignment. Then, the predicted assignment probabilities are used to make adjustments to the data and to then compute the outcome results. The propensity score is used to adjust the distribution of the variables in the groups before examining effects on the outcome. Literature suggests that propensity score analysis has an advantage in accounting for selection bias over regular multivariable regression models (Shah, Laupacis, Hux & Austin, 2005; Sturmer et al., 2006; Weitzen, Lapane, Toledano, Hume, & Mor, 2004). Propensity score analysis adjusts for the difference in the data for groups separately from outcome analysis.

Unlike the multivariable approach, propensity score analysis can be used to create matched data, stratify the data, and to weight the observations in the data. These three different options for utilizing propensity scores is an advantage of the technique compared to the regular multivariable approach. Condensing information into a single score is more useful than including variables as covariates in a regression model, especially in small datasets. The rule of thumb for sample size in fitting a multiple regression model is 50 + 8m, where m = number of independent variables in the model (Tabachnick & Fidell, 2013). The required sample size will increase along with the number of predictors to be included (Tabachnick & Fidell). Therefore, a regular multivariable approach will not be appropriate with large sets of covariates and a small number of observations. In addition, propensity score models can incorporate main effects of variables along with interaction and polynomial terms (non-parsimonious models) in estimating propensity scores. The propensity score models are intended to balance the group and not to make inferences about the comparison groups (D'Agostino, 1998). Thus including interactions and polynomial terms in a propensity score model does not create a problem yet helps to increase precision in finding matches.

Both traditional multivariable and propensity score analysis is only practical in the context of observed biases. Both techniques adjust for the biases from the observed, measured variables; they do not address adjustments for hidden biases in the data. A detailed description and application of the three different propensity score analyses is presented in subsequent sections.

What is propensity score analysis?

Propensity score analysis is a statistical technique that is proposed as the corrective approach in addressing selection bias in quasi-experiments or observational studies (Spreeuwenberg et al., 2010). It was first developed and introduced by Rosenbaum and Rubin from Rubin's causal framework model (Rosenbaum & Rubin, 1983). For more than two decades, propensity score techniques have been used to help in the evaluation of cause-effect analysis in observational studies (Rosenbaum & Rubin, 1983). In terms of statistics, a propensity score is defined as the conditional probability of being assigned to a group in a study given observed characteristics. The probability estimate is calculated based on the observed variable, which might be an intervention or treatment (Spreeuwenberg et al., 2010). The need for propensity scores can be related to sampling procedures. When randomized experiments are not feasible or ethical, the sampling or participant selection into groups becomes potentially dependent on other factors. The non-randomization costs come in potential biases in the samples which lead to biased estimates and misleading interpretations, especially in comparing the between-group effect. A propensity score is a function of a collection of confounding factors contributing to the assignment to treatment. This single estimate, the propensity score, is used to balance confounding variables that differentiate the distribution of the sample into treatment and comparison groups in observational studies (Stone & Tang, 2013). Computation and the use of a propensity score has been found to be effective in reducing bias in observational studies, especially in the presence of a large set of confounding variables (Spreeuwenberg et al., 2010).

A single score from a list of covariates is estimated with the intention of mimicking randomization where the treated and non-treated groups are probabilistically comparable (Harder, Stuart, & Anthony, 2010). When the groups under study (treated and non-treated) are comparable, the distributions of the sample characteristics are probabilistically equivalent across the groups. The similar distribution of the covariates across the groups ensures that there are no prior probabilistic differences between the groups.

Finding an appropriate match between treatment and control cases is essentially the key to the propensity score analysis, where use of propensity scores helps to minimize the differences between the groups and so to minimize biases prior to any statistical analysis. However, matching is not the only option in propensity score analysis as balancing can also be achieved by using the propensity scores as weights and also from a procedure balancing the differences between the groups prior to the outcome analysis. Balancing can also be achieved using stratification by propensity score. All three approaches using propensity score analysis help to adjust for the variability due to observed confounders across the groups. Each of the three propensity score utilization approaches is discussed in the following sections.

Propensity score matching. Matching is an approach used in re-creating samples that are homogenous from the original data. The process of re-sampling using matching helps to eliminate any pre-analysis difference across the groups, thus making them comparable (Rosenbaum & Rubin, 1985). This assures all the potential threats to differential selection have been accounted for and any change in the outcome is a result of the treatment implementation. Matching methods have been used in sociology since the first half of the 20th century (Althauser & Rubin, 1970; Chapin, 1947; Greenwood, 1945) but the theoretical framework for matching methods was developed beginning in the late 1960s and early 1970s. Cochran’s paper in 1968 was the beginning of an introduction of matching along with subclassification. Although Cochran and Rubin worked on the development of matching and stratification, they primarily focused on matching based on a single covariate (Cochran & Rubin, 1973). However, Rubin extended this work by incorporating multiple variables in the matching process (Rubin, 1997).

This work led to the use of propensity scores where the propensity score synthesizes a vector of covariates into a single score that is used to find matches. Matches are sets of individuals from treatment and control groups that have similar characteristics. Each individual from the treatment group is matched to an individual(s) with similar characteristics in the control group. It is challenging to find exact matches for the treated individuals in the presences of a variety of covariates. This is when propensity scores, single scalar values, become useful to find the best matches. Generally, each individual in a treatment group will be randomly matched to an individual(s) from the control group with the closest propensity score. The closest propensity score is determined by calculating the distance/ difference between the propensity score for the selected observations. Distance means the difference between the propensity scores between the matches. The shortest distance or difference indicates closeness between matches.

There are at least nine different types of matching approaches available (Guo & Fraser, 2015). The matching techniques can be categorized into parametric and non-parametric matching. Parametric matching approaches can be further categorized into greedy and optimal matching. The (1) near-neighbor, (2) Mahalanobis distance, (3) caliper matching, and (4) near neighbor with caliper are parametric types of greedy matching. Full and optimal matching belong to the optimal parametric approach. Non-parametric matching includes kernel matching. Optimal matching has an advantage over greedy matching as it optimizes the differences between the overall matches to have the smallest differences (Guo & Fraser, 2015). The max-flow approach of network flow theory tries to optimize the selection of individuals by minimizing the distances between the matched pairs and the overall matches (Guo & Fraser, 2015). Full matching is a type of optimized matching analysis that allows an observation/individual to be matched to many cases (1 to many). Full matching uses all the observations in the data and there are no limits set on the number of matches with individuals from the reference group (Stuart & Green, 2008). Full matching has the advantage of retaining a sufficient number of observations/cases for outcome analysis (Holmes, 2014). In this study, only greedy matching techniques were explored.

In a matching approach, not all the observations are matched. The observations that fail to be matched are excluded from the study. In some cases, the same observation is used more than once (Caliendo & Kopeinig, 2008). If the sample is small, losing cases might lead to power issues in detecting effects. Also, the exclusion of the unmatched sample might have an influence on the studied treatment effect (Caliendo & Kopeinig, 2008).

Stratification using propensity scores. Stratification is also called subclassification (Guo & Fraser, 2015). Stratification is a process of dividing propensity scores into strata (D'Agostino, 1998). Stratification categorizes individuals into relatively homogenous groups. Rosenbaum and Rubin (1984) extended Cochran's stratification idea of categorizing a continuous variable into five subgroups or quintiles. Instead of a continuous variable, Rosenbaum and Rubin categorized observations using propensity scores. Following is a description of propensity score analysis using a stratification approach. First, estimated propensity scores for all the observations in the treatment and control groups are ordered from the smallest to the largest values. Next, the ordered propensity scores are grouped into quintiles. The estimated propensity score is a continuous variable. Strata are generated to slice the propensity score distribution into equally spaced intervals. Each stratum is defined by the range of propensity scores. Individuals within a specified propensity score range are grouped into a specific stratum. Depending on the sample size, each stratum will contain more than two individuals. The number of strata and the range used to create the strata will be the same across the treatment and control groups. Strata can be created either from quintiles, the median, or quartiles.

The choice of number of strata influences the variance and bias in the treatment effect estimate (Myres & Louis, 2012). A larger number of strata produces lower variance and potentially higher bias (Myres & Louis, 2012). Rosenbaum and Rubin (1984) recommended using quintiles as this number of strata was shown to remove 90% of the bias from the data. Therefore, individuals/observations from treatment and control groups that are categorized into the same strata are assumed to have similar characteristics. Cases with a similar propensity score range will be classified into the same stratum. The cases that belong to same stratum are homogenous in their observed characteristics. For two group comparisons, each stratum will contain cases from the treatment and control groups. The average of the mean outcome for treatment and control groups is estimated within each propensity score quintile. The, the difference between the averaged mean outcome for treatment and control groups is estimated for each quintile.

As a result, five difference estimates are computed from the five quintiles. The difference estimates across the five quintiles are summed to estimate the overall treatment effect.

Propensity score weighting. Propensity score weighting is another approach to using propensity scores to account for selection bias. Propensity score weighting is similar to survey sampling that accounts for over- or under-represented samples (Lee, Lessler, & Stuart, 2010). Weights in sampling are generally used to make the distribution of some variables in the data approximate the distribution of those variables in the population (Tabachnick & Fidell, 2013). The distribution of the sample differs from the population because cases may be sampled with unequal probability. For example, over-sampling males will result in a conclusion biased with respect to gender. A propensity score is estimated using the variables that cause different distributions in the treatment and control groups. The weights estimated using propensity scores $(\widehat{e(x)})$ will be used to weight the participants in the treatment and control groups. The weights to estimate the Average Treatment Effect (ATE) for participants in the treatment group are the inverse of the propensity score ((1/$\widehat{e(x)}$). The weights for participants in the control group are the inverse of one minus the propensity score [1/(1-$\widehat{e(x)}$)]. The weights account for the difference in the distribution of the observed covariates between treatment and control groups. Then, the weighted observations are used to estimate the true treatment effect. Propensity score weighting has advantages over matching as it uses all the individuals in control and treatment groups for outcome analysis. The ability to retain all the individuals in estimating the treatment effect helps to maintain statistical power to detect a treatment effect (Stone & Tang, 2013).

The estimated weights can also be used in other univariate or multivariate statistical techniques that allow weights in the analysis. Propensity score weighting has the advantage of being a doubly robust technique. Doubly robust estimation combines inverse probability weighting by propensity score with regression modelling of the relationship between covariates and outcome for each treatment (Robins, Hernan, & Brumback, 2000). Doubly robust estimation correctly estimates as long as either the propensity score model or the outcome regression models are correctly specified; that is, in the absence of unmeasured confounders. This doubly robust estimation allows room for misspecification in the model (Robins et al., 2000). Even if the propensity score model is not well specified and not totally able to remove a confounding effect, the regression model will account for it thus making the treatment effect estimates unbiased. However, there are some limitations in the use of propensity score weighting. Freedman and Berk (2008) noted propensity scores as sensitive to a mis-specified propensity score model and will have an impact on treatment effect estimation. Propensity score weights from mis-specified models can exert a negative effect on the treatment effect estimate (Harder, Stuart, & Anthony, 2010). Besides, a mis-specified propensity score model could potentially yield extreme weights for the observations and lead to potentially biased estimates (Lee et al., 2010).

Propensity score analysis with multiple treatment groups

Propensity score analysis can be extended to treatment variables that are continuous or have multiple categories (Guo & Fraser, 2015). The application of propensity score analysis in multiple treatment group treatment variables is an extension of the framework created by Rosenbaum and Rubin (1983). Increasing demand for studies with continuous values on treatment variables in the medical and social sciences fields fueled the need to expand propensity score application to multiple treatment groups. Literature relevant to the investigation of propensity score applications with more than two treatment groups or with a continuous treatment variable is limited. However, within the limited literature, researchers applied propensity scores in multiple groups in two directions. The two directions are (1) use of a single scalar balancing score, and (2) generalized propensity score techniques. Following is a description of propensity score analysis directions in a multiple treatment group context.

First direction: Single scalar balancing score.

The use of an ordinal logistic regression model in estimating a propensity score is key to the application of a single scalar balancing score (Guo & Fraser, 2015). A single scalar score is applicable when the values of the treatment variable are ordered. Here, ordinal logistic regression uses cumulative probability to predict the likelihood of being in one category versus all lower or higher categories (Hosmer & Lemeshow, 2000). Following is an illustration of how a single scalar score for propensity score is computed and used in propensity score analysis. First, ordinal logistic regression is used to compute a single scalar score using equation 2:

$$\text{Log }\left(\frac{\Pr(Z_k \geq d)}{\Pr(Z_k < d)}\right) = \theta_d + \beta_k' \quad \text{for d = 2, 3} \tag{2}$$

Where k = distribution of level of category for sample of participants given the observed covariates (β_k'). The model compares the probability of a response or treatment category (Z_k) greater than or equal to a given category (d = 2, 3) to the probability of a response less than this category (d = 1). The model simultaneously estimates multiple logit equations to estimate the log of the odds. The number of equations estimated is the number of categories in the dependent variable minus one. Each of the equations estimates the odds being in a category over the lower categories. For each category, a set of coefficients for the observed covariates are observed. The vector of covariates together with the coefficients (β_k') specific to the treatment variable are used to quantify the single scalar score called a propensity score.

Next, the estimated propensity score is used to compute the distance between the participants. The following equation is used to compute the distance between the observations for multiple treatment conditions. The smallest difference between the observations is computed. Observations with the smallest difference are selected as the matched pairs.

$$\Delta\ (X_k\ , X_{k'}) = \frac{(\widehat{\beta}' X_k - \widehat{\beta}' X_k') + \epsilon}{(Z_k - Z_{k'}\ )^2} \tag{3}$$

where X_k ,$X_{k'}$ are the pair of observations that will be compared. $\widehat{\beta}'X_k,\ \hat{\beta}'X_k'$ are the propensity scores for the pairs that are being compared. The membership of the pairs to be compared are represented by $Z_k,\ Z_{k'}$. Finally, ϵ represents the random error term.

Lu et al. (2011) illustrated the steps taken in performing optimal non-bipartite matching using single scalar propensity scores. Lu et al. also created the nbpMatching package in R to perform optimal non-bipartite matching analysis. The goal is to find matches for observations with the smallest difference in their observed propensity scores. The computed distances between the participants are transferred into a distance matrix. Then, pairs with the smallest distance are selected. The optimal non-bipartite matching algorithm is used to determine the minimal total distance between the matched pairs in addition to the distance within the paired matches. The optimal non-bipartite matching allows the smallest distance within and between the matches.

This approach gained less attention in the field as it requires a complex algorithm to create a matched sample. An accepted algorithm in solving non-bipartite matching is by searching augmenting paths (Papadimitriou & Steiglitz, 1982). Lu et al. (2011) proposed and implemented Derig's shortest augmentation path algorithm to solve the non-bipartite matching problem. However, to date there is no literature available presenting the application of single scalar propensity score technique in an applied context.

Stratification using single scalar propensity scores. Zanutto, Lu, and Hornik (2005) extended the stratification approach to a multiple treatment group context. The authors evaluated the impact of the National Anti-Drug Media Campaign on the level of alcohol and drug usage. Their study illustrated steps in performing stratification in the context of multiple propensity scores. First, the propensity score was estimated using an ordinal logit model. Then, the extreme propensity scores in each therapy group were excluded by trimming the data. The trimmed propensity scores for each therapy were stratified into five quintiles. The balance on covariates in each therapy group within each stratum was examined. The quintiles of propensity scores and treatment levels were used as factors in a two-way ANOVA which were tested on each continuous covariate to examine the group differences. Logistic regressions were used to assess the balance in binary covariates. Once the covariates are balanced, the outcome for each therapy group is averaged and added together for an overall treatment effect.

Second direction: Generalized propensity scores

The generalized propensity score is the second direction of propensity score application in multi-treatment groups. Imbens (2000) proposed a generalized propensity score technique which estimates multiple propensity scores through multinomial logit models. This approach is computationally more straightforward than single scalar balancing scores. First, a generalized propensity score is estimated using a multinomial logit model. The number of generated propensity scores is dependent on the number of treatment categories. For example, each participant in the data will have three propensity scores if there are three levels of treatment. Multinomial logistic regression in propensity score analysis is used to predict the probabilities of the different possible outcomes of a categorically distributed treatment variable given a set of independent variables (Hosmer & Lemeshow, 2000).

The multinomial logistic regression model simultaneously estimates binary logits for all possible comparisons among the outcome categories (Hosmer & Lemeshow, 2000). With three treatment outcomes, multinomial logistic regression is equivalent to performing three binary logistic regressions comparing treatment groups 1 to 2, 1 to 3, and 2 to 3. For example, the probability of being in group 1 versus group 2 can be expressed as the following.

$$\text{Ln}\,[\frac{\Pr(1\,|x)}{\Pr(\,2\,|x)} = \beta_{0,\ A|B} + \beta_{1,\ A|B}] \tag{4}$$

For a 1 versus 2 paired comparisons, the binary logit model means the probability of being in group 1 over 2 is a function of a linear combination of variables.

The use of generalized propensity scores

Literature shows that generalized propensity scores can be utilized in matching, weighting, and covariate adjustments. Following is an example illustrating steps in applying multiple propensity scores in the context of covariate adjustment, weighting, and matching.

Generalized propensity scores as covariates in a regression model.

Spreeuwenberg et al. (2010) studied the impact of five therapies on the severity of psychiatric symptomology. Their study presented a step by step guide in using multiple propensity scores as covariates in a regression model in the context of mental health.

They suggested the following steps in the application of multiple propensity scores in covariate regression adjustments. First, Spreeuwenberg et al. recommend estimating the treatment effect using statistical analysis before any corrections. Next, group differences on the observed variables are examined using analysis of covariance (ANCOVA) for continuous and logistic regressions for categorical variables. Variables for a propensity score model are selected and used to compute the propensity score estimates. A multinomial logistic regression model is developed with the entire set of selected variables related to treatment assignment (outcome) used as the independent variables and treatment group membership as the dependent variable. Using a particular therapy as the reference category, four propensity scores are estimated. Next, the overlap of the propensity score distribution is examined. Overlap is examined to ensure that each subject in the study has a certain probability of having been assigned to one of the rest of the therapy groups.

Next, balance in the observed covariates is checked after correcting for the biases. The same therapy that was used as the reference category earlier is used as the reference category in creating dummy variables for the therapy group. Finally, a continuous outcome is regressed on multiple propensity scores (ps), dummy coded treatment groups (d), and the product of dummy therapy and propensity scores (dxps) using multiple regression. The coefficient estimates for the dummy variables present the difference between the therapy and reference therapy group that is referred to as treatment effect. Spreeuwenberg et al. (2010) suggested this approach as relatively straightforward compared to other propensity score techniques such as matching and stratification. This is because finding matches from five treatment groups is complex and difficult, therefore Spreeuwenberg et al. (2010) proposed using a generalized propensity score as a covariate in a regression model as a practical approach in the presence of multiple treatment groups.

Weighting using generalized propensity scores. Multiple propensity scores from multiple treatment groups can be also applied in a weighting approach. McCaffrey et al. (2013) illustrated steps in performing weighting using multiple propensity scores. Following are the steps proposed by McCaffrey et al. in generating and applying propensity scores as weights for the group with multiple treatment exposures. They studied the impact of three different treatments on the level of alcohol and drug usage. First, binary generalized boosted models (GBM) (the GBM is described below) are fitted for each of the three treatment groups. Balance in the observed covariates is assessed using absolute standardized bias. Also, Kolmogorov-Smirnov statistics were used to assess the balance on the observed covariates.

Propensity scores that are estimated respective to the treatment group are used as the weights in estimating the treatment effect. For example, treatment groups 1 and 2 are compared. Here, the propensity scores for group 1 and 2 will only be used to weight the observations from treatments 1 and 2. The difference between the weighted group 1 and 2 outcome is one of the three average treatment effects that can be computed from the study. The toolkit for the non-equivalent group (twang) package in R is used to compute the propensity score weights (McCaffrey et al., 2013). Then, a survey package is used to compute the treatment effect (Lumley, 2014). However, the twang package only uses generalized boosted models (GBM) to estimate the propensity scores. According to McCaffrey, Ridgeway and Moral (2004), GBM is a data-mining technique that has proved considerable success in predicting treatment assignment using a large number of covariates. GBM relies on regression trees using an iterative algorithm to estimate the function that describes the relationship between a set of covariates and the treatment variable (Li, 2012). Through the iterative process, the function estimating treatment assignment includes interactions and polynomial terms to produce a better function estimating the treatment variable. In twang, the number of regression trees is specified to indicate the number of iterations and the best function predicting treatment assignment is determined using the balance criteria. The function that achieves the best balance in the covariates across the comparison groups is determined as the best propensity score model. In twang, the mean difference and Kolmogorov-Smirnov test are used to assess the balance between groups. The potential implication of using GBM is predicting treatment assignment is that the propensity score estimate is subject to change according to the GBM specifications. Changes in the GBM models in terms of iterations and balance assessments such as Kolmogorov-Smironov approach could change the propensity score estimations. Also, estimation using GBM models does not specify the prediction model used in estimating the probability and makes it difficult to replicate the predictive model. For example, the polynomial and interaction terms used for predicting propensity score cannot be identified when using GBM.

Matching using generalized propensity scores. Soberay (2015) applied propensity score matching using multiple propensity score estimates. This study was an application of multi-group propensity score analysis to study the outcomes related to treatments for problem gambling. The TriMatch package in R was used to find triplet matched pairs for three treatment groups (Bryer, 2013). The package includes matching with and without replacement and with and without calipers. In the TriMatch package, the minimum distances between the matched observations are evaluated and not the overall distance with all the matched triplets. This package uses a greedy matching approach as opposed to optimal matching approach in finding matches. Following are the steps illustrated by Bryer (2013) in applying propensity score matching in multiple treatment groups.

First, multinomial logistic regression is performed to estimate multiple propensity scores. Then, the group with the smallest sample size is selected and set as a reference group (e.g., group 1). The first observation from the reference group is selected. An observation from group 2 with the smallest distance from a selected observation from the reference group is selected. Subsequently, the observation from group 3 with the smallest distance compared to selected observations from group 2 is identified. The distance of the selected observation from group 3 is compared to the selected observation from the reference group.

This process is repeated for each observation in the treatment group until matched triplets are created. Then, repeated measures ANOVA is used to examine differences in the treatment effect between the groups. A matched t-test is proposed as a post-hoc analysis to specifically estimate the mean difference across treatments (average treatment effect). The limitation of TriMatch algorithm is that it only works for studies with three levels of treatment. This limits the application of this package when the number of treatment groups is greater than three.

Summary of propensity score analysis in multiple treatment group

The reviewed literature on multiple group propensity score analysis introduces steps in applying propensity scores in multiple treatment group groups. Generally, the application of propensity scores in multiple groups is similar to propensity score application in binary conditions. The selection of variables, modelling the selection model, balance checking, and propensity score adjustment techniques (matching, stratification, and weighting) are similar between binary and multiple treatment groups. The difference is that the steps in propensity score analysis are repeated for all potential pairwise comparisons in the multiple treatment groups. It is crucial to clearly define the pairwise comparisons since this affects the selection of reference groups in computing propensity score estimates. Overall, propensity score analysis in multiple groups is just an extension of the binary application of propensity scores except that it becomes complex in the presence of multiple treatment groups.

Chapter Two: Method

Introduction

Chapter Two includes a detailed description of the study’s methodology. In this chapter, I have listed the steps that I used to answer my research questions. First, I describe Monte Carlo simulation under research design. Next, I explain three phases of my analysis procedures which were (1) data generation, (2) propensity score analysis, and (3) outcome analysis. The data generation phase includes generation of variables. Subsequently, the generated variables were manipulated and combined into a dataset. Scenarios were created to illustrate the conditions in the data. Data with specific conditions were tested using three different propensity techniques. Finally, the performance of each propensity score technique was evaluated under outcome analysis.

Design

This study involved a Monte Carlo (MC) simulation. Monte Carlo simulation is an empirical method for generating data for the purpose of evaluating the performance of statistics. MC methods are used in an experiment where data are generated to test specific theoretically derived hypotheses (Paxton, Curran, Bollen, Kirby, & Chen, 2001). In MC simulation, a statistical distribution is identified and used as the source for each of the input parameters. Then, random samples from each distribution are drawn which represent the values of the input variables. Monte Carlo simulation allows researchers to assess the finite sampling performance of estimators by creating controlled conditions from which sampling distributions of parameter estimates are produced. Knowledge of the sampling distribution is the key to the evaluation of the behavior of a statistic (Paxton et al., 2001). For example, a researcher can determine the bias of a statistic from the sampling distribution, as well as its efficiency and other desirable properties. Sampling distributions are theoretical and unobserved. However, the MC method is used to create simulated data reflecting the sampling distribution. In this study, the variables for a synthetic dataset were generated with known population parameters. Then, I drew repeated samples of size N from the population. Subsequently, statistical analysis was performed on the generated datasets to address the research questions.

Analysis procedure

The analysis procedure of this study consisted of three main phases. The three phases were (1) data generation, (2) propensity score analysis, and (3) outcome analysis. There were multiple steps within the three phases. Following is the detailed description of the steps within the three phases.

Phase 1: Data generation

Phase 1 included (1) variable generation, (2) treatment variable computation, (3) outcome variable computation, (4) specification of average treatment effect, and (5) creating scenarios for data conditions. Following is a detailed description of all five steps within the data generation phase.

Variables generation. In the first phase, data for the study were artificially generated using Monte Carlo computer simulation. A total of six variables was used in data generation. The six variables (x1-x6) were treated as independent, continuous, and normally distributed variables.

Variable selection is a crucial issue faced by researchers estimating propensity scores (Brookhart et al., 2006). Rubin and Thomas (1996) suggested including all variables thought to be related to the outcome despite being related to exposure or not. Later Rubin (1997) suggested including variables that are strongly related to exposure but not necessarily related to the outcome. In order to account for both opinions, this study included variables that were treatment assignment specific, outcome specific, and both treatment and outcome specific in estimating propensity scores. For this study, the role of the independent variables in the treatment and outcome variable was predetermined using Austin, Grootendorst, and Anderson’s (2007) variable association matrix.

The matrix of variable associations was used as a guiding framework to specify variables that was used to estimate the treatment and outcome variables. Following is the matrix illustrating the variables and their relationship to treatment and outcome. The 3 x 3 matrix is the degree of relationship between independent variables to treatment and outcome variables. The row in the matrix represents the strength of relationship of independent variables to the treatment variable. The column in the matrix represents the strength of the relationship of the independent variable to the outcome variable. Table 2 below illustrates the variable association matrix.

Table 2

Variables and their Relation to Treatment and Outcome

	Treatment		
Outcome	Strongly associated with treatment	Moderately associated with treatment	Not associated with treatment
Strongly associate with outcome	x1	x2	x3
Moderately associated with outcome	x4	x5	x6
Not associated with outcome	x7	x8	x9

The association matrix helped to set up the models to explain the role and strength of the independent variables estimating treatment and outcome variables. Based on this framework, six variables were generated. Two variables (x1-x2) were specified for predicting the treatment variable. Another two variables (x5-x6) were assigned to specifically compute the outcome variable. Variables x3 and x4 were used in estimating both treatment and outcome variable. Table 3 summarizes the role and strength of all six variables in predicting the treatment and outcome variables. The association between independent, treatment, and outcome variables were used to decide coefficients for regression models estimating treatment and outcome variables. In the next step, the coefficients for the variables in treatment assignment and outcome regression models were specified.

Table 3

Variables and their Relation to Treatment and Outcome in this Study

Variables	Treatment	Outcome
x1	Strong association	No association
x2	Moderate association	No association
x3	Moderate association	Moderate association
x4	Moderate association	Moderate association
x5	No association	Strong association
x6	No association	Moderate association

Computing the treatment variable. A nominal treatment variable with three groups of treatment exposure was created. Four independent variables (x1, x2, x3, and x4) were used to generate the treatment variable. As mentioned earlier, the variables were set to have different strength in the assignment of treatment levels. The strength of the variables was determined using correlation coefficients. Here, the value of the coefficient was estimated from Hinkle, Wiersma, and Jurs' (2003) table of correlation coefficients (Table 4). Following is the step taken in generating a nominal treatment variable with three groups.

Table 4

Size of the Correlation Coefficient and the Interpretation

Size of Correlation	*Interpretation*
.90 to 1.00 (-.90 to –1.00)	Very high positive (negative) correlation
.70 to .90 (-.70 to -.90)	High positive (negative) correlation
.50 to .70 (-.50 to -.70)	Moderate positive (negative) correlation
.30 to .50 (-.30 to -.50)	Low positive (negative) correlation
.00 to .30 (.00 to -.30)	Little if any correlation

The treatment variable was generated from a multinomial distribution with varying treatment group probabilities. The probabilities of three treatment groups were computed in the following steps.

Step 1: A selection equation was defined. The selection equation was used to model the relationship between the independent variables and treatment assignments (groups). The relationship (coefficient) of each independent variable to each treatment group was modelled. Then, the exponential value of the estimate was computed for each treatment group. In the exponential equations, coefficients for all independent variables were set to differ across the three groups. The coefficient values were set to be different to generate unequal observations across groups. As a result, the proportion of observations in control, treatment 1, and treatment 2 were set to be different. The control group (C1) was designed to have the highest number of observations and followed by treatment 1 (T1) and treatment 2 (T2) (equation 5-7). Following is the equation estimating treatment assignment for all three groups.

$$C1 = \exp[-0.20(x_1) + (-0.70(x_2)) + (-1.7(x_3)) + (-2.0(x_4)) + \varepsilon\ ] \quad (5)$$

$$T1 = \exp[-0.30(x_1) + (-0.65(x_2)) + (-1.6(x_3)) + (-2.1(x_4)) + \varepsilon\ ] \quad (6)$$

$$T2 = \exp[-0.35(x_1) + (-0.50(x_2)) + (-1.5(x_3)) + (-2.2(x_4)) + \varepsilon\ ] \quad (7)$$

The probability of the treatment assignment for each group was computed in step 2. Estimated probability (ProbL1) was the likelihood of an observation/participant being in a specific treatment group over other two groups (equation 8-10).

$$ProbL1 = C1/(C1 + T1 + T2) \quad (8)$$

$$ProbL2 = T1/(C1 + T1 + T2) \quad (9)$$

$$ProbL3 = T2/(C1 + T1 + T2) \quad (10)$$

In step 3, the computed probabilities (ProbL1, ProbL2, and ProbL3) were included in the multinomial distribution function to generate the treatment variable. As a result, the generated treatment variable consisted of three levels. The three levels were (1) control, (2) treatment 1, and (3) treatment 2.

Computing the outcome variable. Independent variables x3 to x6 were used to compute a continuous outcome variable. Appropriate coefficient estimates reflecting the relationship between independent and outcome variables was specified using a correlation coefficient table (Table 4). The coefficient values for the independent variables were randomly selected from the given correlation range. The previously generated nominal treatment variable was spilt into two dummy variables (t1, t2). Dummy coded variables that indicated the group membership were used along with other independent variables to estimate the outcome. Following is the regression equation illustrating the coefficients used to estimate the outcome variable.

$$Y = [\ 0.3(x_3) +\ 0.4(x_4) + 0.95(x_5) + 0.6(x_6) + 5(t1) + 15(t2) +\ \varepsilon] \qquad (11)$$

These generated variables were combined and stored in a dataset for further analysis.

Specifying average treatment effect (ATE). Average treatment effect is the mean (µ) difference between two groups. In three groups, the control group was set as the reference category. With control group as the reference category, two ATEs were estimated. In this study, the mean of the treatment effect for treatment 1, 2, and control was set to 10, 20, and 5. The first ATE (ATE 1) estimate was the mean difference between treatment 1 and control. Here, ATE 1 was 5 (10-5). The observations in treatment group 1 have outcome variable with 5 points higher than the outcome observations in control group Next, the ATE (ATE 2) for treatment group 2 versus control was 15 (20-5). Here, the observations in treatment group 2 were set to have value of the outcomes 15 points higher than the outcome observations in the control group. In both treatment groups, the values for the outcomes were set to be higher than in control group. This is to indicate the large effect of the implementation of the treatments in the outcome. Both the ATE values were set at 5 and 10 based on pilot testing. Therefore, the values were set higher to observe the changes after propensity score adjustments. Table 5 presents the formulae for estimating ATE and their true mean estimates defined prior to data generation and analysis.

Table 5

True Treatment Effect Estimate and Estimation Formula

Effect	Formula for ATE	True ATE
Treatment 1 vs Control (ATE 1)	μ_{t1} - μ_{ct}	5
Treatment 2vs Control (ATE 2)	μ_{t2} - μ_{ct}	15

Creating data. Data were simulated with the following structured conditions. Following are the two scenarios created to compare the performance of three propensity score techniques. The two scenarios imposed overt and hidden biases on the data. For each scenario, the correlation between the treatment, outcome, and error terms in both treatment and outcome variables were manipulated. First, the overt bias scenario (scenario 1) is explained and is followed by the hidden biases scenario (scenario 2). For better understanding, I present a causal model which is used to illustrate the association between the variables (Figure 1). The model is used to explain the conditions for overt and hidden biases. Following is the model and scenarios for the study.

General model for the scenario

The model consists of a treatment variable, outcome variable, independent variables, and error terms associated with both treatment and outcome variables. The treatment variable for each group was measured using the following regression equation

$$T \ = \ [\beta_1(x_1) + \ \beta_2(x_2) + \beta_3(x_3) + \beta_4 7(x_4) + v] \qquad (12)$$

Four independent variables which were x1, x2, x3, and x4 were regressed on the treatment variable. The error term (v) refers to the amount of unexplained variance in estimating the treatment effect after accounting for four independent variables.

Next, the outcome variable was measured using regression equation 8. Here, the treatment variable was dummy coded and used in computing the outcome variable. The residual term (u) means the unexplained variance in the outcome after accounting for six predictors.

$$Y = [\beta_1(x_3) + \beta_2(x_4) + \beta_3(x_5) + \beta_4 7(x_6) + \beta_5(t_1) + \beta_6(t_2) + u] \quad (13)$$

Both the regression equations were combined into a single model as illustrated in Figure 1. The purpose was similar to path analysis, where it simultaneously examines a set of relationships between one or more variables (Ullman, 1996). In this study, path analysis was conducted to examine the causal relationship between the treatment and outcome variables after controlling for all the independent variables. Figure 1 illustrates the relationship between the treatment and outcome variables. In Figure 1, the treatment variable is presented as a single nominal variable with three levels.

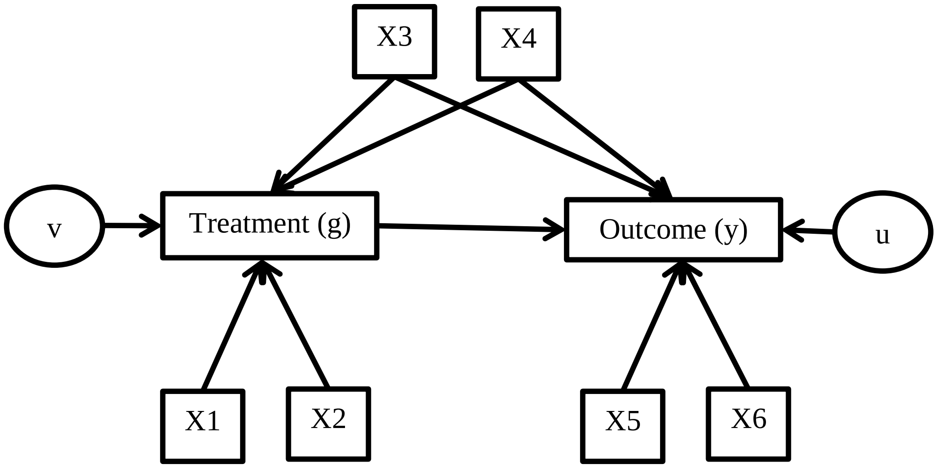

Figure 1

Basic Model Illustrating the Relationship between the Treatment and Outcome Variables

Scenario 1: Overt bias. Overt bias is bias that can be seen in the data at hand; for instance, bias that can be seen prior to treatment (Rosenbaum, 2002). Overt bias can be observed and measured. Overt bias is a type of bias that is usually controlled using statistical adjustments such as matching and stratification. In this scenario, the general model was modified to create scenario 1. Model 1 for this scenario consisted of a treatment (three levels), independent variables (x1-x6), outcome variable, and two error terms. Treatment was predicted by variable x1, x2, x3, and x4. The error term (v) for treatment and outcome (u) was generated. With these variables, this scenario assumed the following conditions. First, the explanatory variables (x1-x6) were set to be uncorrelated.

The association between independent variables was set to range from 0.1 to 0.2 to indicate independence among variables (Hinkle et al., 2003).

Second, the correlations between independent variables (x1-x6) and treatment error (v) were set to zero. This ensures that the treatment variable is correctly measured and no variables are omitted in computing the treatment variable. Next, the correlation between treatment error (v) and outcome error (u) was be set to be uncorrelated. The error term for treatment and outcome variables reflects unexplained variability in the variables after accounting for the independent variables. When the two errors are independent, it assures no hidden biases in estimating treatment variables. Fourthly, the means of the treatment assignment variables (x1-x4) were set to be different across treatment and control groups. This was to establish imbalances in the covariate distribution across treatment and control groups.

Finally, variables (x1 and x2) were set to be correlated with the outcome error (u). The specified correlation indicates the omitted variable that should be modelled along with other variables in predicting the outcome. The error term for the outcome (u) is the residual variance that is left to be explained after accounting for treatment and x3-x6 variables. Since x1and x2 contributes to treatment assignment, failing to account for these variables in predicting the outcome will impose large residuals and biased estimates. Therefore, the correlation between the two independent variables (x1, x2) and error term for outcome (u) imposes overt bias in the dataset. All five steps used in generating the data for the overt bias condition are described in the R scripts attached under Appendix A. Figure 2 illustrates the relationship between the variables for scenario 1. The correlation imposing overt bias is highlighted in the dotted line in Figure 2. Table 6 presents the summary of the conditions imposed for scenario 1.

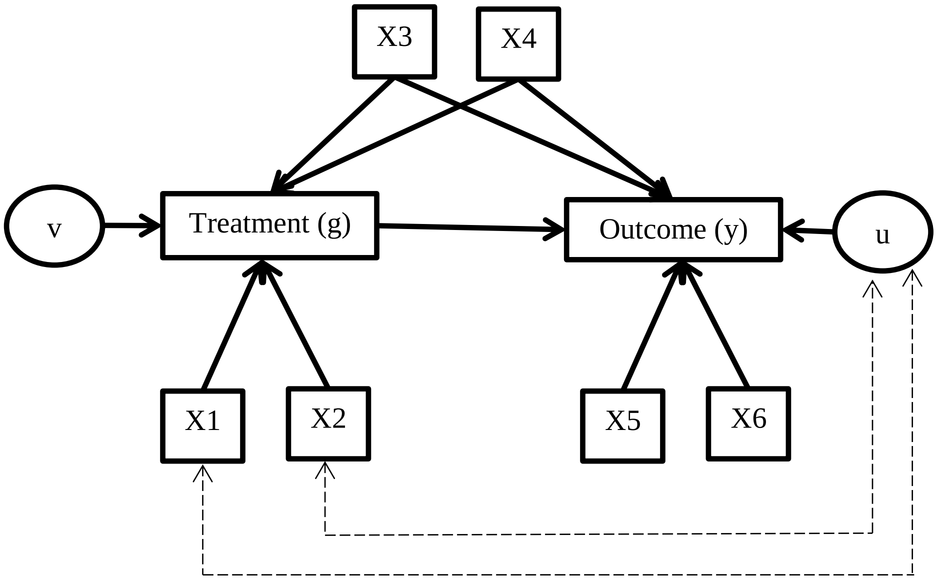

Figure 2.

Model for Scenario 1 (Overt Bias) Illustrating the Relationship Between the Treatment and Outcome Variables

Table 6

Summary of the Conditions Set for Scenario 1

No	Condition	Explanation
1	All six independent (x1-x6) variables was set to be significantly uncorrelated.	This is to control for potential variability that will be introduced by correlations in the data.
2	Treatment assignment variables (x1-x4) were independent of the treatment variables error term (v).	Error term assures that treatment variable is accurately specified by the four variables.
3	Both treatment (v) and outcome error (u) terms were set to be uncorrelated.	This ensured that no unmeasured variables in treatment assignment were omitted in estimating the outcome.
4	Means of (x1-x4) variables were set to be different across the treatments and control group.	The purpose was to install imbalances in the covariate distribution across treatments and control groups.
5	Variables that predict outcomes (x3-x6) except for x1 and x2 were set independent of the outcome variable’s error (u) term.	Independent variables that were related to treatment assignment were omitted when estimating the outcome. Variables were omitted to impose overt bias in the dataset.

For scenario 1, the following correlation matrix between the variables was specified. The correlation among the independent variables was set in the range of .10 to .20. The range was proposed as it reflects little or no relationship between the variables when generating data for the overt bias scenario. Next, the correlation between the treatment assignment variables (x1, x2) with the error term of outcome (v) model was set to uncorrelated. The same matrix was used, except that the correlation estimates between the two treatment assignment variables (x1 and x2) with outcome error term (u) was set at .30, .50 and .70 to represent low, moderate, and high correlation. Three correlation matrices were used to generate data with different levels of overt bias. Within the three levels of overt bias, 200, 500 and 1000 observations were generated. This was to study the influence of sample size in overt bias. In total, nine datasets (3 correlation levels x 3 sample sizes) with overt bias were generated. Table 7 is an example of a correlation matrix for the small sample size (n=200) with a low level of overt bias (r=0.3). The correlation imposing a low level of correlation is indicated in bold-face in the correlation matrix (Table 7).

Table 7

Correlation Matrix for the Small Level of Overt Bias

	x1	x2	x3	x4	x5	x6	u	V
x1	1	0.15	0.12	0.1	0	0	**0.30**	0
x2	0.15	1	0.10	0.12	0	0	**0.30**	0
x3	0.12	0.10	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	**0.30**	**0.30**	0	0	0	0	1	0
V	0	0	0	0	0	0	0	1

Scenario 2: Hidden bias. Source bias in a selection process that cannot be modelled or observed directly is called hidden bias (Rosenbaum, 2002). Since the bias is unmeasurable, it cannot be corrected for in propensity score analysis. This becomes a

drawback in the application of propensity scores in correcting for biases. Hidden bias was modelled under scenario two. Similar to scenario one, the outcome variable was predicted by four independent variables (x1-x4) and a treatment variable (t). The following conditions were set in creating the second scenario. First, the relationship between the explanatory variables was specified. Here, the correlations between the explanatory variables (x1-x6) were set to a low correlation, 0.10 to 0.20. The relationship between the variables was controlled to reduce potential variability in the generated data.

Second, the correlations between independent variables (x1-x6) with treatment (v) and outcome (u) error terms were set to zero. This ensured that both treatment and outcome variables were correctly measured without omitting any independent variables. Next, the means of the treatment assignment variables (x1-x4) was set to be different across treatment and control groups. This was to establish imbalances in the covariate distribution across treatment and control groups. Finally, the treatment (v) and outcome (u) errors were set to be correlated. The unexplained variability in the treatment and outcome variables after accounting for the independent variables refers to error terms. When the error terms were set to be correlated, it imposed hidden bias in the data. This is because the correlation between errors means that there are unmeasured or unidentified potential factors influencing treatment assignment that will result in biased estimates. The association between treatment (v) and outcome (u) error terms was set at .30, .50 and .70 to represent low, moderate, and high correlations. The correlation imposing hidden bias is highlighted in the dotted line in Figure 3. The summary of the conditions defined for scenario 2 (Table 8) and the respective correlation matrix (Table 9) to generate data for scenario 2 are described after Figure 3.

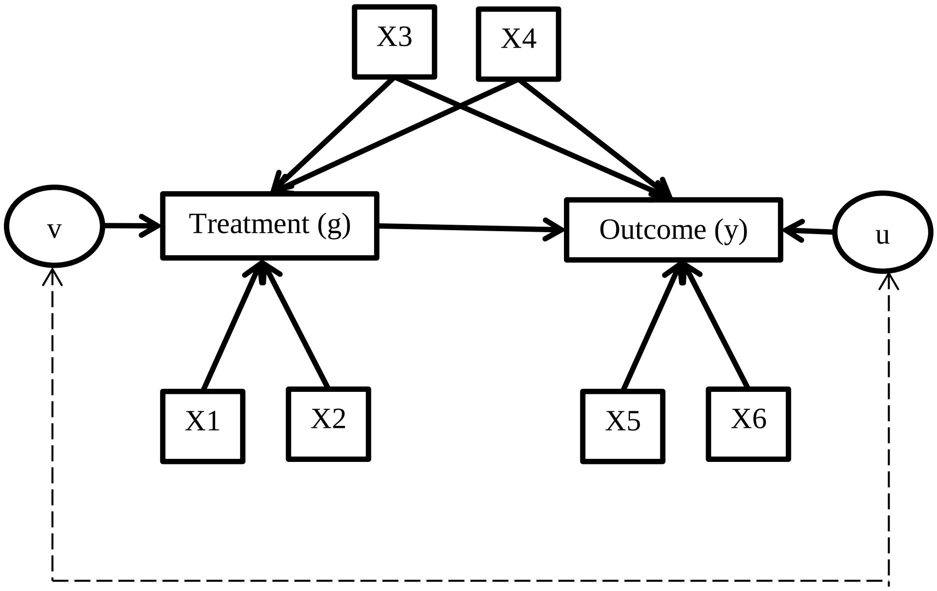

Figure 3.

Model for Scenario 2 (Hidden Bias) Illustrating the Relationship between Treatment and Outcome Variables

Table 8

Summary of the Conditions Set for Scenario 2

No	Condition	Explanation
1	All six independent variables (x1-x6) were set significantly uncorrelated.	This was to control for potential variability that will be introduced by correlations in the data.
2	Treatment assignment variables (x1-x4) were independent of the treatment variables error term (v).	Error term assured that treatment variable was accurately specified by the four variables.
3	Variables that predict outcomes (x3-x6) was set to be independent of the outcome variable's error term (u).	This was to specify that there were no variables related to treatment assignment omitted in estimating the outcome.
4	Means of (x1-x4) variables were set to be different across the treatments and control group.	The purpose was to install imbalances in the covariate distribution across treatments and control groups.
5	Both treatment (v) and outcome (u) error terms were set to be correlated.	This ensured that there were unmeasured variables in treatment assignment omitted in estimating the outcome.

For scenario 2, the following correlation matrix between the variables was specified. The correlations between the independent variables were in the range between 0.10 and 0.20. The range was proposed to show little or no relationship between the variables when generating data for the hidden bias scenario. Next, the correlation between the treatment assignment variables (x1-x4) with the error term of the outcome (u) model was set to be zero. The correlation between error terms for treatment (u) and outcome (v) was set at .30, .60 and .90 to represent small, moderate, and high correlation. Three correlation matrices were used to generate data with different levels of hidden bias. Within the three levels of hidden bias, 200, 500, and 1000 observations were generated. This was to study the influence of sample size in hidden bias. In total, nine datasets (3 correlations x 3 sample sizes) with hidden bias were generated. Table 9 is an example of the correlation matrix for a low level of hidden bias. The correlation imposing low hidden bias is indicated in bold-face in the correlation matrix (Table 9).

Table 9

Correlation Matrix for the Small Level of Hidden Bias

	x1	x2	x3	x4	x5	x6	u	V
x1	1	0.15	0.12	0.1	0	0	0	0
x2	0.15	1	0.1	0.12	0	0	0	0
x3	0.12	0.1	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	0	0	0	0	0	0	1	**0.3**
V	0	0	0	0	0	0	**0.3**	1

Phase 2: Propensity score analysis

Three propensity score techniques were used to account for bias in the data. Propensity score matching and weighting were performed using the TriMatch, and twang packages in R. To date, there is no available package in R for performing stratification. Therefore, I wrote a function in R for stratification analysis. Following is a description of each propensity score technique.

Matching. Propensity score matching was performed using TriMatch (Bryer, 2013). Here, I describe the steps in TriMatch to obtain matched observations. First, the propensity score was estimated using three separate logistic regressions. In this study, three logistic regressions were conducted to estimate the likelihood of being in group 1 versus 2, group 1 versus 3, and group 1 versus 2. With three logistic regressions, each subject in the study had three propensity scores which were saved to the case record. The difference between the participant's propensity score in the first group and the participant's propensity score in groups 2 and 3 was computed. The difference was saved as the distance. The first observation from group 1 was selected. An observation from group 2 with the smallest distance from selected observation 1 was selected. Subsequently, an observation from group 3 with the smallest distance compared to selected observations from group 2 was identified. The distance of the selected observation from group 3 was compared to the selected observation from group 1. Observations with a distance less than a .25 caliper size were retained. The observations that met the minimal distance requirement were saved as matched triplets to the database.

Stratification. Stratification is a process of dividing propensity scores into strata (Guo & Fraser, 2015). Individuals/observations with similar characteristics are grouped together into strata. There is to date no package in R that is able to run stratification for treatment with more than two levels. Therefore, a function for the R software was created as part of the data analysis. Zanutto, Lu, and Hornik's (2005) recommended steps in performing stratification in multiple treatment group was based on a single scalar approach. However, this study focused on the generalized propensity score approach. In this study, the application of a stratification technique for two groups was extended to three groups. Following are the steps taken in conducting stratification in three groups. First, the observations that were assigned to treatment 1 and control groups were extracted and assembled into a group (group 1). Then, a propensity score was estimated for the participants in group 1 through logistic regression. The predicted probabilities indicating treatment group membership from logistic regression for group 1 were saved as the propensity scores. The propensity score for treatment 1 and control groups was stratified into five quintiles. As a result, the observations in treatment 1 and control were categorized into five strata. The observations from treatment 1 and control that were assigned to the same stratum share the same propensity score value. The same procedure was repeated for observations assigned to treatment 2. For treatment 2, only the cases that were assigned to treatment 2 and control were used for stratification (group 2). At the end, two sets of data were created from the original dataset. In both group 1 and 2, the treatment and control observations were categorized into strata. The stratification approach helped to organize the data into strata prior to outcome analysis.

Weighting. Propensity score weighting is another bias corrective technique under the general area of propensity score analysis. A propensity score is estimated and used as a weight to obtain a balanced sample (Imbens, 2000). The twang package in R was used to perform propensity score weighting. Following are the proposed steps in estimating propensity score weights for three treatment groups (McCaffrey et al., 2013). First, a treatment group was selected as the reference group. In this study, there were three levels of treatment (C, T1, and T2); control group (C) was set as the reference category. Next, data that only contained individuals from the reference group (C) and individuals in group (T1) were extracted. The subset dataset contained individuals from group (C) and (T1). The propensity score $\tilde{p}_1(X_i)$ was computed using a Generalized Boosted Model (GBM) for the individuals as the probability of being in treatment exposure (T1) versus (C). Then, the estimated propensity scores were transformed into odds ratios (OR) using equation 11:

$$OR = \frac{\tilde{p}_1(X_i)}{(1-\tilde{p}_1(X_i))} \tag{14}$$

The odds ratio for the last category (T2) was set to 1. Then, the computed odds ratio was transformed back to a probability (propensity score) so that they are on the same scale using Equation 12:

$$\widehat{p}_t\,(X_i) = \frac{OR_{it}}{\sum_{j=1}^{M} OR_{ij}} \tag{15}$$

where OR_{it} is odds ratio for ith individual in t group and $\sum_{j=1}^{M} OR_{ij}$ is the total odds ratio from all three groups. The computed probability was used as the weight in estimating the outcome. The Toolkit for non-equivalent group (twang) package in R is available for performing propensity score weighting for multiple treatment groups. However, the package only uses a Generalized Boosted Regression Model to estimate propensity scores. This has been identified as one of the limitations of this study.

Phase 3: Outcome analysis

Average Treatment Effect (ATE) was computed for each dataset after applying propensity score techniques. ATE is the mean outcome difference between two comparison groups. In this study, two ATE's were computed from two pairwise comparisons. Two pairwise comparisons are treatment 1 versus control, and treatment 2 versus control. The mean outcome differences within the pairwise comparisons were estimated as the average treatment effects. Following are the steps in estimating ATE after matching, stratification, and weighting approaches.

Matching. Repeated measures ANOVA was performed on the matched data. Then, post-hoc analysis was performed across two pairwise comparisons to estimate the mean differences. The two pairwise comparisons were (1) treatment 1 vs control, and (2) treatment 2 vs control. For each pairwise comparison, the difference between the true and estimated average treatment effect (ATE) was computed using Equation 16

$$\text{Bias} = \text{Computed treatment effect} - \text{True treatment effect} \tag{16}$$

True average treatment effect is one of parameters that was pre-determined when generating data in phase 1. The amount of bias from the true ATE for all two comparisons was computed and compared across propensity score techniques.

Stratification. In stratification, the difference between the mean outcomes of the treatment and control group was estimated for each stratum. Then, the difference between the mean outcomes of treatment and control was averaged across all five quintiles. For example in group 1, the difference between the mean outcome of treatment 1 and mean outcome of control group was estimated within each quintiles 1-5. The differences across the quintiles were averaged to represent the treatment effect of the group. The treatment effect of the groups was the ATE. Two ATE's were computed and they were comparisons between the averaged outcome of treatment 1 versus control, and treatment

2 versus control. Then, the difference between the true and estimated ATEs was computed using Equation 13. The amount of bias from the true ATE for both the comparisons was computed to compare across propensity score techniques.

Weighting. A weighted regression on treatment was performed to assess the effect on the outcome. The control group was set as the reference category in creating two dummy variables. The outcome was regressed on two dummy coded treatment variables to estimate the treatment effect. The estimated coefficient for dummy treatment variable 1 (β_1) represents the ATE for treatment 1 relative to control. The coefficient from the second dummy treatment variable (β_2) represents the ATE for treatment 2 relative to control. These computed ATE's were compared to the true ATEs using equation 13 to estimate the bias. The computed bias estimations were saved to make comparisons across propensity score techniques.

Summary of the Analysis

Within each scenario, 27 (3 levels of bias x 3 sample sizes x 3 propensity score techniques) datasets were generated. Therefore, for overt and hidden bias scenarios, a total of 54 data conditions were generated to assess the ATE. Next, each of the 54 data conditions was replicated 1000 times. The difference between the true and estimated ATE for the 1000 replications within each condition was calculated and stored. For each of the 27 conditions under overt bias, the difference between true and estimated treatment effect for each replication was indexed into Excel files. These 27 files were reformatted by including the information regarding conditions in addition to the difference between true and estimated treatment effect values. The 27 data files with a total of 27000 (27 x 1000) observations was merged into a single file for further analysis.

Analysis of variance (ANOVA) was used to evaluate the performance of the three propensity score techniques under different data conditions. Two 3x3x3x2 analyses of variance were conducted to evaluate the effects of propensity score technique (technique), level of bias (level), sample size (sample size), and type of treatment effect (effect) on the amount of bias in estimating the treatment effect under overt and hidden bias conditions. The main and interaction effects of the four factors on the amount of bias in estimating the treatment effect were examined. All the analyses were performed using R statistical software. The Statistical Package for the Social Sciences (SPSS) package was used to generate plots. Figure 4 is a visual representation of the analysis procedure.

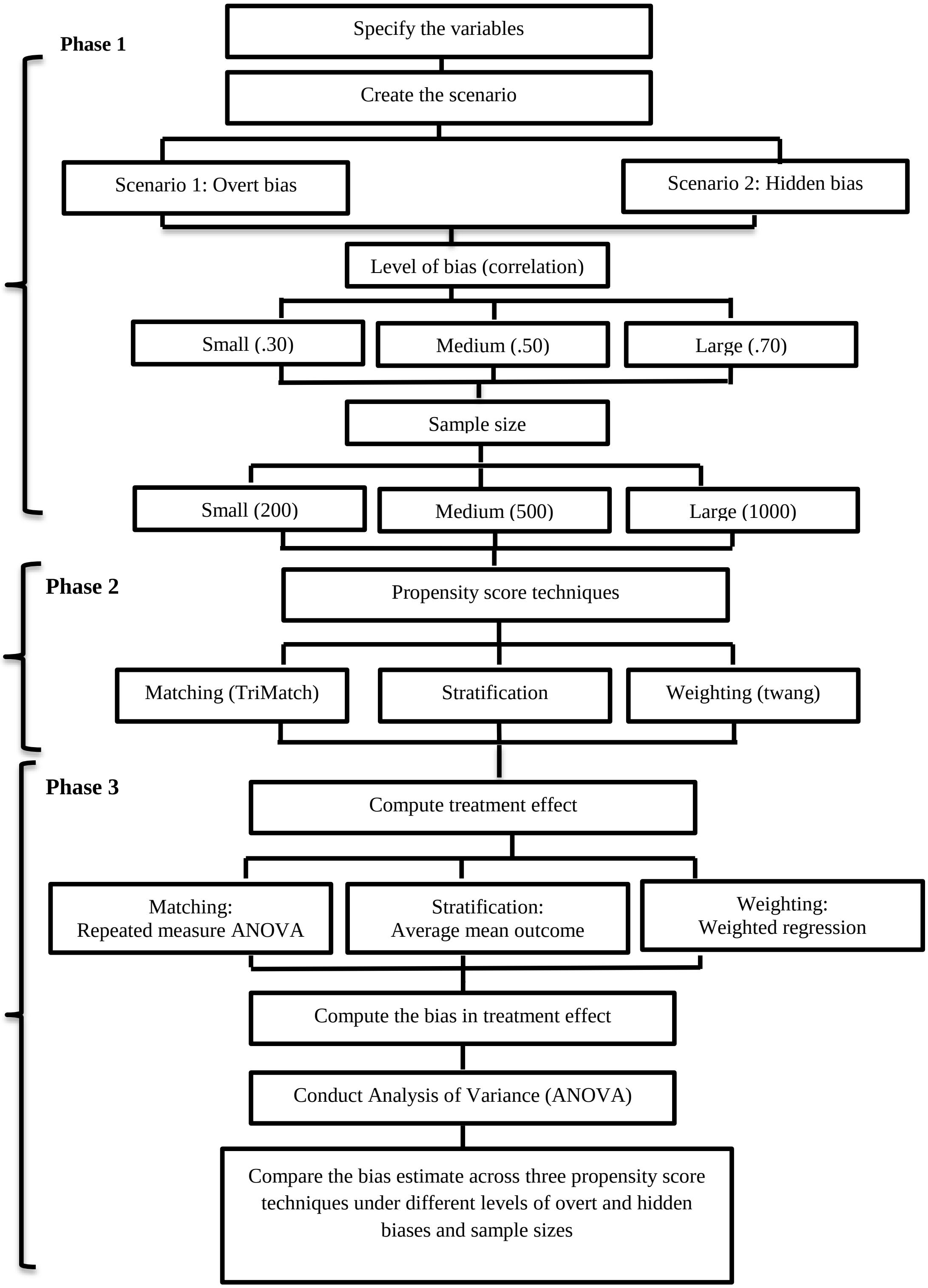
Phase 1
Specify the variables
Create the scenario
Scenario 1: Overt bias
Scenario 2: Hidden bias
Level of bias (correlation)
Small (.30)
Medium (.50)
Large (.70)
Sample size
Small (200)
Medium (500)
Large (1000)
Phase 2
Propensity score techniques
Matching (TriMatch)
Stratification
Weighting (twang)
Phase 3
Compute treatment effect
Matching: Repeated measure ANOVA
Stratification: Average mean outcome
Weighting: Weighted regression
Compute the bias in treatment effect
Conduct Analysis of Variance (ANOVA)
Compare the bias estimate across three propensity score techniques under different levels of overt and hidden biases and sample sizes

Figure 4
Visual Representation of the Analysis Procedure

Chapter Three: Results

Introduction

The ultimate goal of this study was to determine the propensity score technique that performs the best under different conditions. The success of a propensity score technique addressing selection biases is determined through the estimated treatment effects. Treatment effect (effect size) is the difference between the means of treatment and control group outcomes. In this study, two effect sizes were computed and they were (1) difference between treatment 1 and control, and (2) difference between treatment 2 and control. The estimated effect sizes were compared to the pre-defined true effect sizes. In this simulation, the true treatment effect between treatment 1 and control was set to be 5 and the difference between treatment 2 and control was set at 15. The differences between the computed effect size 1 and 5, and effect size 2 and 15 were computed as the amount of bias. If the amount of bias is zero, then the true and estimated effect sizes are the same. Thus, there is no bias in estimating the treatment effect. If the amount of bias is lower than zero (negative estimate) then the estimated treatment effect size is lower than the true effect size. A lower value indicates an underestimation of the treatment effect. The treatment effect is considered overestimated if the difference between the estimated and true treatment effect size (amount of bias) is greater than zero.

The chapter begins with an overview of the results and proceeds to a detailed description of the findings. This chapter is organized into two sections and they are results related to (1) descriptive and (2) inferential analyses. The descriptive analysis section includes an overall summary of the amount of bias found under the overt and the hidden bias condition. In the inferential analysis section, a detailed description of 4-way analyses of variance (ANOVA) is presented. The description of the 4-way ANOVAs, assumption checking, and the findings answering the research questions are presented for both overt and hidden bias under the inferential section.

There was a large number of observations in both the overt and hidden bias conditions. The large number of observations (n=27,000) increased the power of detecting the smallest effect in the data. Therefore, effect size was computed to determine the significant contribution of each factor. Cohen’s rule of thumb specific for partial eta squared were used. Cohen’s rule of thumb for small, medium, and large partial eta-square was set at .10, .25, and .40 (Cohen, 1992). Partial eta-squared (effect size) of at least .10 was needed for the factor to be considered significant instead of relying on the p-value (Hutchinson & Olmos, 1998). The partial eta-squared was used to assess the main and interaction effects from overt and hidden bias conditions. The evidence for the research questions is presented in the narrative and in tables. The findings were followed up with overall conclusions for both overt and hidden bias. The chapter concludes with the similarities and differences between the overt and hidden bias findings.

Section 1: Descriptive Analysis

The difference between the true and estimated treatment effect size (amount of bias) was computed for each replication. The mean of the amount of bias for 1000 replications for overt and hidden biases conditions was estimated. In Table 1, the mean amount of bias is presented for each level within technique, level of bias, sample size, and type of treatment effect size factors for overt and hidden bias. Overall, the treatment effect size estimates after propensity score adjustments were found to be lower than the true treatment effect size estimates. The amount of bias in estimating the treatment effect was found to be lower in the presence of hidden bias compared to overt bias conditions. The negative direction in the amount of bias for all four factors indicates that the treatment effect was underestimated after the selection bias adjustment (Table 10). Two 4-way ANOVAs were conducted to further investigate the difference in the amount of bias within the levels of each factor and the interactions between the factors for overt and hidden bias conditions.

Table 10

Summary of the Amount of Bias by Technique, Level of Overt Bias, Sample Size and Treatment Effect

	Overt bias			Hidden bias		
Factor with levels	**N**	**Mean**	**SD**	**N**	**Mean**	**SD**
Technique						
Propensity score matching	18000	-1.06	0.33	18000	-0.52	0.30
Propensity score stratification	18000	-1.19	0.84	18000	-0.72	0.84
Propensity score weighting	18000	-0.75	0.23	18000	-0.40	0.19
Level of bias						
Small (r =.3)	18000	-0.82	0.55	18000	-0.54	0.54
Medium (r =.5)	18000	-1.00	0.55	18000	-0.55	0.54
Large (r = .7)	18000	-1.18	0.55	18000	-0.55	0.54
Sample size						
Small (n = 200)	18000	-1.14	0.89	18000	-0.71	0.87
Medium (n = 500)	18000	-0.93	0.31	18000	-0.47	0.25
Large (n = 1000)	18000	-0.93	0.23	18000	-0.46	0.14
Type of treatment effect						
Treatment effect 1	18000	-0.95	0.33	18000	-0.48	0.26
Treatment effect 2	18000	-1.05	0.73	18000	-0.61	0.71

Section 2: Inferential Analysis

The inferential analysis section includes results from four-way ANOVAs for the overt and for the hidden bias conditions. First, the findings under the overt bias condition are discussed. Next, the results under the hidden bias condition are presented.

Overt bias. Following is a description of the four-way ANOVA, assumptions, results, and the summary of the findings for the overt bias condition.

Description of the four-way ANOVA. A 3x3x3x2 analysis of variance was conducted to evaluate the effects of propensity score technique (technique), level of overt bias (level), sample size (sample size), and type of treatment effect (effect) on the amount of bias in estimating the treatment effect. A between-subjects factorial design was employed. The technique factor had three levels and they were (1) propensity score matching, (2) stratification using propensity scores, and (3) propensity score weighting. The three levels which comprised level of overt bias were (1) small, (2) medium, and (3) large. The sample size factor comprised three levels and they were (1) small, (2) medium, and (3) large. Lastly, the type of treatment effect consisted of (1) mean difference between treatment 1 and control (treatment effect 1), and (2) mean difference between treatment 2 and control (treatment effect 2).

Assumption checking. While the independence assumption was met, a statistically significant violation of homogeneity of variance was found for technique, level of overt bias, sample size, and type of treatment effect. Normality was violated with skewness for the stratification technique and small and medium levels of overt bias. However, analysis of variance is robust with respect to violations of homogeneity of variance and normality with a sufficient number of cases and a balanced design. In this study, there were a large and equal number of cases in each cell.

Findings from the four-way ANOVA. The main effects, two, three, and four-way interactions from ANOVA were used to examine if there was any difference due to propensity score technique, sample size, and level of overt bias in the amount of bias found in estimating the treatment effect. Following is the summary of the four-way ANOVA. Results from the ANOVA were divided and organized by the level of interactions between the factors. The omnibus results from the main and interaction effects are reported with suggestions for follow up analysis (Table 11). Table 11 presents the interaction and main effects of the four factors along with the empirical findings and is followed by interpretation of the findings.

Table 11

Results from 4-way ANOVA with its Interpretations under Overt Bias Condition

4-way and 3-way Interactions for Overt Bias Condition

Effect	**Results**	**Interpretation and action**
Technique x Sample Size x Overt bias x Type	*F*(8, 53946) = 0.05, *p*>.99, η^2 =.0	• The four-way interaction had no impact on estimating the treatment effect
Technique x Sample Size x Overt bias	*F*(8, 53946) = 1.00, *p* =.43, η^2 =.0	• All three way-interactions between the independent variables had no substantial effect on the treatment effect estimation.
Technique x Overt bias x Type	*F*(4, 53946) = 0.11, *p* =.98, η^2 =.0	
Technique x Sample Size x Type	*F*(4, 53946) = 938.65, *p* < .001, η^2 =.07	
Sample Size x Overt bias x Type	*F*(4, 53946) = 0.05, *p* >.99, η^2 =.0	

2-way Interactions for Overt Bias Condition

Technique x Sample Size	*F*(4, 53946) = 1710.16, *p* < .001, η^2 =.11	• The difference between propensity score technique in estimating the treatment effect was affected by the sample size. • The interaction was further investigated using simple effects and pairwise comparisons.
Technique x Overt bias	*F*(4, 53946) = 45.67, *p* < .001, η^2 =.0	• These two-way interactions between
Technique x Type	*F*(2, 53946) = 1061.20, *p* < .001, η^2 =.04	

Sample x Overt bias	$F(4, 53946) = 2.11$, $p = .08$, $\eta^2 =.0$	the independent variables had negligible impact on the treatment effect estimation.
Overt bias x Type	$F(2, 53946) = 1.98$, $p = 0.14$, $\eta^2 =.0$	
Sample x Type	$F(2, 53946) = 1013.48$, $p < .001$, $\eta^2 =.04$	

Main Effects for Hidden Bias Condition

Technique	$F(2, 53946) = 4671.06$, $p < .001$, $\eta^2 =.15$	• Technique had a small effect on the treatment effect estimation. • No follow up of the main effect was conducted due to the presence of technique by sample size interaction. • The interaction implies that the difference between the techniques differs by sample size.
Sample Size	$F(2, 53946) = 1352.58$, $p < .001$, $\eta^2 =.05$	• Sample size had no substantial effect on the treatment effect estimation
Overt bias	$F(2, 53946) = 2984.62$, $p < .001$, $\eta^2 =.10$	• Level of overt bias had a small effect on the treatment effect estimation. • Pairwise comparisons between levels of overt bias were explored.
Type	$F(1, 53946) = 779.22$, $p < .001$, $\eta^2 =.01$	• There was no difference in the amount of bias estimated in treatment effect 1 and treatment effect 2

Key Points from ANOVA Results for the Overt Bias Condition

The four-way and three-way interactions had effect sizes substantially less than .10 and were considered negligible (Table 11). In addition, the two-way interactions between the variables also had negligible effects on the amount of bias in estimating treatment effect except for the technique by sample size interaction (η^2 = .11). Only the main effect of overt bias, (η^2 = .10) was found to have a substantial effect on the difference between the estimated and true treatment effect. Figure 1 is a visual representation of the results and follow up analysis. Following Figure 1 is the description of the follow up analysis for main and interaction effects from the four-way ANOVA. First, a pairwise comparison between the levels of overt bias is discussed as the follow up analysis for the significant main effect of overt bias. Subsequently, the results of a simple effects analysis and pairwise comparison were reported as the follow up to assess the significant interaction between propensity score technique and sample size.

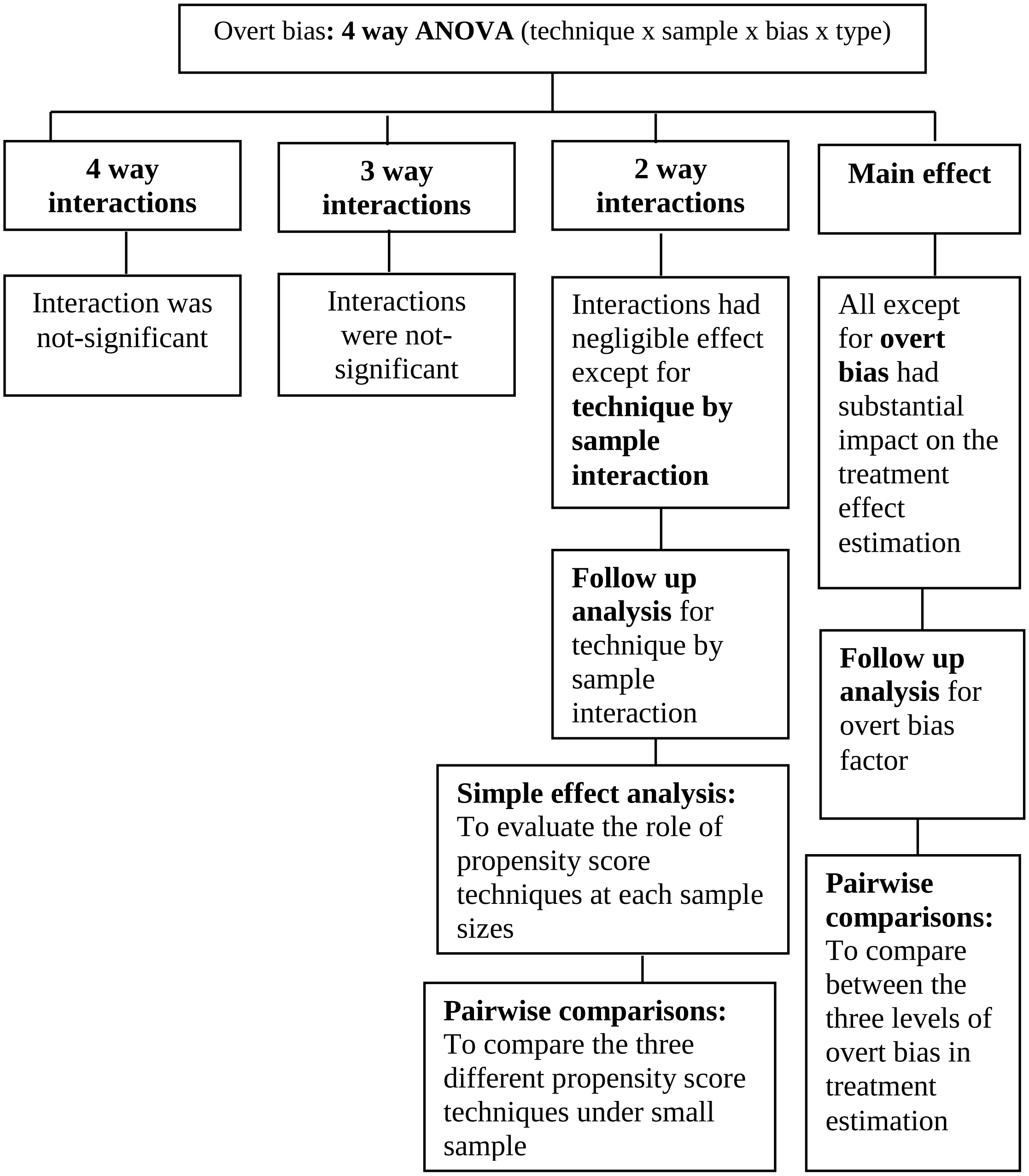
Overt bias: 4 way ANOVA (technique x sample x bias x type)
4 way interactions
3 way interactions
2 way interactions
Main effect
Interaction was not-significant
Interactions were not-significant
Interactions had negligible effect except for technique by sample interaction
All except for overt bias had substantial impact on the treatment effect estimation
Follow up analysis for technique by sample interaction
Follow up analysis for overt bias factor
Simple effect analysis: To evaluate the role of propensity score techniques at each sample sizes
Pairwise comparisons: To compare between the three levels of overt bias in treatment estimation
Pairwise comparisons: To compare the three different propensity score techniques under small sample

Figure 5

Visual Representation of the Results and Follow Up Analysis for Overt Bias Condition

Follow Up Analysis for the Main Effect

Pairwise comparisons between levels of overt bias. The level of overt bias factor had a small effect ($\eta^2 = .10$) on the amount of bias in estimating the treatment effect (Table 11). A Tukey post-hoc analysis with family-wise error correction was conducted at $\alpha = .05$. The pairwise results show that there was a difference in the amount of bias estimated in the presence of overt bias. The amount of bias between large versus small overt bias was twice as large compared to medium versus small levels of overt bias (Table 12). Similarly, the amount of bias between large versus medium was half as large compared to large versus small levels of overt bias. The condition with a small level of overt bias had treatment effect estimates that were closer to the true values (Table 12).

Table 12

Mean, Standard Deviation, Difference and Confidence Interval Values for Levels of Overt Bias Factor

	Mean	**SD**	
Small	-0.82	0.55	
Medium	-1.00	0.55	
Large	-1.18	0.55	
		95% CI	
Difference		**LB**	**UB**
Medium vs Small	-0.18	-0.19	-0.17
Large vs Small	-0.36	-0.37	-0.35
Large vs Medium	-0.18	-0.19	-0.17

Note: LB = Lower bound, UB = Upper bound

Follow Up Analysis for the Interaction Effect

Interaction effect between technique and sample size. The two way interactions between the variables had negligible effects on the amount of bias in estimating treatment effect except for the technique by sample size interaction (η^2 = .11), (Table 11). The interaction suggests the performance of propensity score techniques differ by sample size. Figure 2 presents the interaction between technique and sample. The figure supports the existence of differences in the treatment effect estimation between propensity scores technique by sample size. The average amount of bias in estimating the treatment effect was different between the three techniques under the small sample size condition (Table 13). On the other hand, the three propensity score analysis techniques performed similarly in the medium and large sample size conditions. Simple effects analyses and pairwise comparisons were performed to explore the technique by sample interaction. The simple effects analysis was performed to assess the effect of propensity score techniques within each level of sample size (small, medium, and large).

Table 13

Means and Standard Deviation of Amount of Bias by Propensity Score Techniques and Sample Size

	Sample size					
	Small (n =200)		**Medium (n = 500)**		**Large (n =1000)**	
Technique	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
Matching	-1.07	0.43	-1.05	0.3	-1.05	0.25
Stratification	-1.65	1.29	-0.98	0.34	-0.95	0.19
Weighting	-0.70	0.28	-0.76	0.2	-0.79	0.18

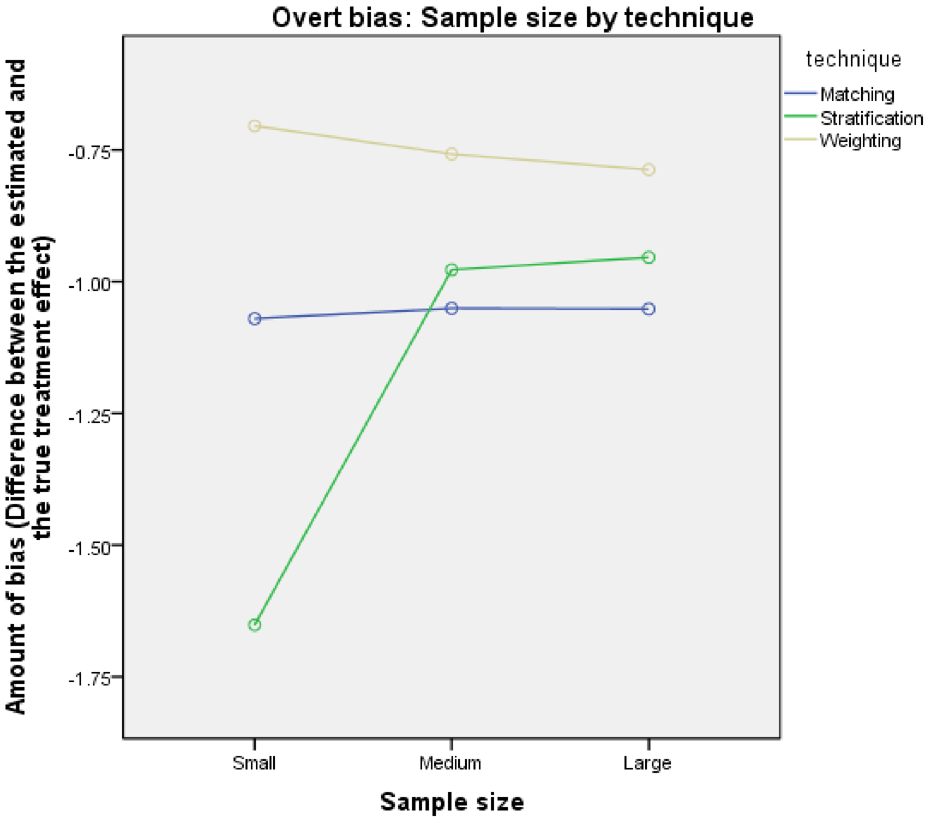

Figure 6

Interaction between Propensity Score Technique and Sample Size in Overt Bias

Simple Effects Analysis on Technique by Sample Interaction

The propensity score technique was found to have an impact for the small sample size condition, η^2 =.17. The propensity score technique had a negligible effect for the medium and large sample size conditions. On the other hand, results showed that the three propensity score techniques performed differently under the small sample size level (Table 14). The difference between three propensity score techniques for the small sample size condition was explored using pairwise comparisons. The pairwise comparisons were performed to address the role of sample size between propensity score techniques.

Table 14

Summary of Simple Effect Analysis on Technique by Sample Size Interaction

Source	**SS**	**df**	***F***	***p-value***	**Effect size (η^2)**
Small sample	2736.9	2	5380.52	< .001	.17
Medium sample	278.5	2	547.53	< .001	.02
Large sample	214.4	2	421.46	< .001	.02
Within	13732.0	53991			

Pairwise Comparisons as a Follow Up to the Simple Effects Analysis

In the small sample size condition, there was a difference in the amount of bias between stratification using propensity scores and propensity score weighting techniques (η^2 =.16). On the other hand, there were no substantial differences between propensity score matching and propensity score weighting and between propensity score matching and stratification using propensity scores in estimating the treatment effect (Table 15). The amount of bias in estimating the treatment effect was lower for stratification using propensity scores (M = -1.65, SD = 1.29) compared to propensity score weighting (M =-.70, SD = .28) (Table 13). Result shows that stratification using propensity scores underestimated the treatment effect the most compared to matching and weighting approaches under the small sample size condition.

Table 15

Summary of Pairwise Comparison Analysis of Propensity Score Technique in Small Sample

Source	**SS**	**Df**	***F***	***p-value***	**Effect size**(η^2)
Matching-Stratification	1013.27	1	3983.95	< .001	.07
Matching-Weighting	401.56	1	1578.83	< .001	.03
Stratification-Weighting	2690.58	1	10578.76	< .001	.16
Within	13731.98	53991			

Summary for overt bias

For overt bias, all three propensity score analysis techniques underestimated the treatment effect. The treatment effect estimates after correcting for selection biases were lower than the true treatment effect. The three propensity score techniques were found to perform differently in the small sample size condition. The stratification technique was found to underestimate the treatment effect the most compared to the matching and weighting approach. The treatment effect estimate was closer to the true treatment effect when using the propensity score weighting adjustment approach. Results showed that the level of overt bias does influence the amount of bias in treatment estimation. The treatment effect estimates were closer to the true effect in the presences of small overt bias. Propensity score techniques were found to be insensitive to the presence of small, medium, and large levels of overt bias in the data.

Section 2: Inferential Analysis

Hidden bias. Following is the description of the four-way ANOVA, assumptions, results, and the summary of the findings for hidden bias condition. In this section, the impact of the four factors on the amount of bias in estimating treatment effect was examined on the data for the hidden bias condition.

Description of the four-way ANOVA. A 3x3x3x2 analysis of variance was conducted to evaluate the effects of propensity score technique (technique), level of hidden bias (level), sample size (sample size), and type of treatment effect (type) on the amount of bias in estimating the treatment effect. A between-subjects factorial design was employed. The technique factor had three levels and they were (1) propensity score matching, (2) stratification using propensity scores, and (3) propensity score weighting. The three levels which comprised level of hidden bias were (1) small, (2) medium, and (3) large. The sample size factor comprised three levels and they were (1) small, (2) medium, and (3) large. Lastly, the type of treatment effect consisted of (1) mean difference between treatment 1 and control (treatment effect 1), and (2) mean difference between treatment 2 and control (treatment effect 2).

Assumption checking. While the independence assumption was met, a statistically significant violation of homogeneity of variance was found for technique, level of overt bias, sample size, and type of treatment effect. Normality was violated with skewness of stratification technique and small and medium levels of hidden bias.

Findings from the four-way ANOVA. The main effects, two, three, and four-way interactions from ANOVA were used to examine if there was any influence of propensity score technique, sample sizes, and level of hidden bias in the amount of bias in estimating the treatment effect. Following is the summary of the four-way ANOVA. Results from the ANOVA were divided and organized by the level of interaction between the factors. The omnibus results from the main and interaction effects are reported with suggestions for follow up analysis (Table 16).

Table 16

Results from 4-way ANOVA with its Interpretations under Hidden Bias Condition

4-way and 3-way Interactions for Hidden Bias Condition

Effect	**Results**	**Interpretation and action**
Technique x Sample x Overt bias x Type	*F*(8, 53946) = 0, *p* >.99, η^2 =.0	- The four-way interaction had no impact on estimating treatment effect
Technique x Sample x Hidden bias	*F*(8, 53946) = 0.20, *p* >.99, η^2 =.0	- All three way-interactions between the independent variables had negligible effect on the treatment effect estimation.
Technique x Hidden bias x Type	*F*(4, 53946) = 0, p >.99, η^2 =.0	
Technique x Sample x Type	*F*(4, 53946) = 960.93, *p* < .001, η^2 =.07	
Sample x Hidden bias x Type	*F*(4, 53946) = 0, *p* >.99, η^2 =.0	

2-way Interactions for Hidden Bias Condition

Technique x Sample	*F*(4, 53946) = 1591.62, *p* < .001, η^2 =.11	- The difference between propensity score techniques in estimating treatment effect was affected by the sample size. - The interaction was further investigated using simple effect and pairwise comparisons.

Technique x Overt bias	*F*(4, 53946) = 45.67, *p* < .001, η^2 =.0	• These two-way interactions between the independent variables had no substantial impact on the amount of bias in treatment estimation.
Technique x Type	*F*(2, 53946) = 1061.20, *p* < .001, η^2 =.04	
Sample x Overt bias	*F*(4, 53946) = 2.11, *p* = .08, η^2 =.0	
Overt bias x Type	*F*(2, 53946) = 1.98, *p* = 0.14, η^2 =.0	
Sample x Type	*F*(2, 53946) = 1013.48, *p* < .001, η^2 =.04	

Main Effects for Hidden Bias Condition

Technique	F(2, 53946) = 2231.96, p < .001, η^2 =.08	• Techniques had negligible impact on the amount of bias in estimating treatment effect
Sample	F(2, 53946) = 1731.64, p < .001, η^2 =.06	• Sample had no substantial effect on the amount of bias in treatment estimation
Hidden bias	F(2, 53946) = 0.20, p = .82, η^2 =.0	• Level of overt bias had no substantial effect on the treatment estimation.
Type	F(1, 53946) = 1104.97, p < .001, η^2 =.02	• There was no considerable difference in the amount of bias estimated in treatment effect 1 and treatment effect 2

Key Points from ANOVA on Hidden Bias Condition

All the four-way and three-way interactions between the independent variables (technique, sample size, level of hidden bias, type of effect size) had negligible effect on the treatment effect estimate. All two-way interactions between the independent variables reported no effect except for the interaction between technique and sample size, η^2 =.11 (Table 16). The interaction suggests the performance of propensity score technique under hidden bias is subject to sample size condition. In addition, the main effects of the four factors were negligible. Figure 3 illustrates the summary of the findings with the follow up analysis. Following Figure 3 is the results from follow up analysis for the small interaction effect between technique and sample size. A simple effect and pairwise comparison were reported as the follow up to assess the significant interaction between propensity score technique and sample size.

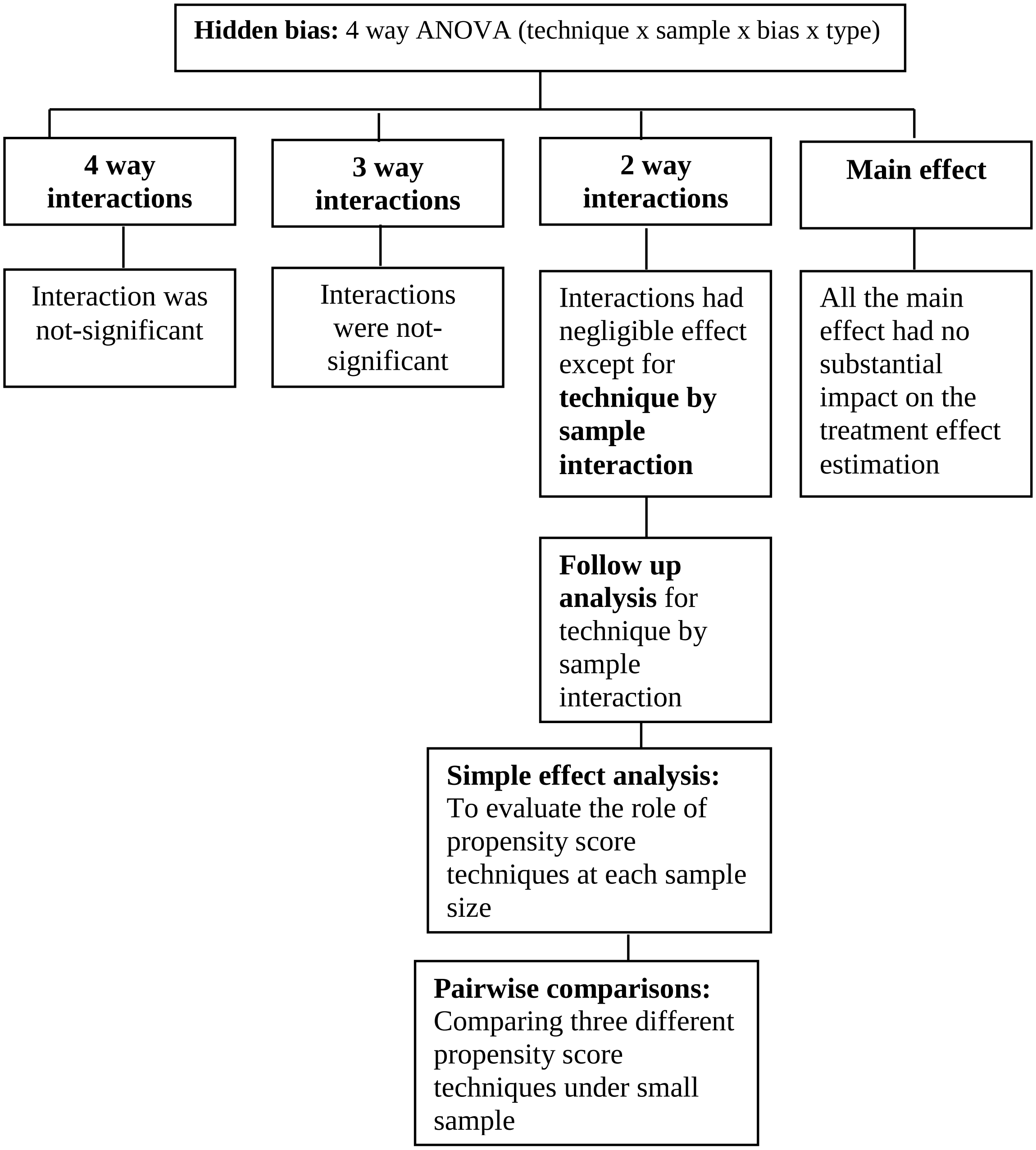
Hidden bias: 4 way ANOVA (technique x sample x bias x type)
4 way interactions
3 way interactions
2 way interactions
Main effect
Interaction was not-significant
Interactions were not-significant
Interactions had negligible effect except for **technique by sample interaction**
All the main effect had no substantial impact on the treatment effect estimation
Follow up analysis for technique by sample interaction
Simple effect analysis: To evaluate the role of propensity score techniques at each sample size
Pairwise comparisons: Comparing three different propensity score techniques under small sample

Figure 7

Visual Representation of the Results and Follow Up Analysis for Hidden Bias Condition

Follow Up Analysis for the Interaction Effect

Interaction between technique and sample size**.** The two-way interaction between the variables had negligible effects on the amount of bias in estimating the treatment effect except for the technique by sample size interaction (η^2 = .11) (Table 16). The interaction suggests the performance of propensity score techniques differed by sample size. Figure 4 presents the interaction between technique and sample. The figure supports the presence of differences between propensity scores technique by sample size. Simple effects analyses and pairwise comparisons were performed to explore the technique by sample interaction. The simple effects analysis was performed to assess the effect of propensity score techniques within each level of sample size (small, medium, and large). Next, pairwise comparisons were performed to assess the difference between the propensity scores in the small sample size condition.

Table 17

Means and Standard Deviation of Amount of Bias by Propensity Score Techniques and Sample Size in Hidden Bias

	Sample size					
	Small (n =200)		**Medium (n = 500)**		**Large (n =1000)**	
Technique	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
Matching	-.53	.42	-.51	.26	-.51	.18
Stratification	-1.19	1.29	-.49	.31	-.47	.11
Weighting	-.70	.28	-.40	.16	-.40	.11

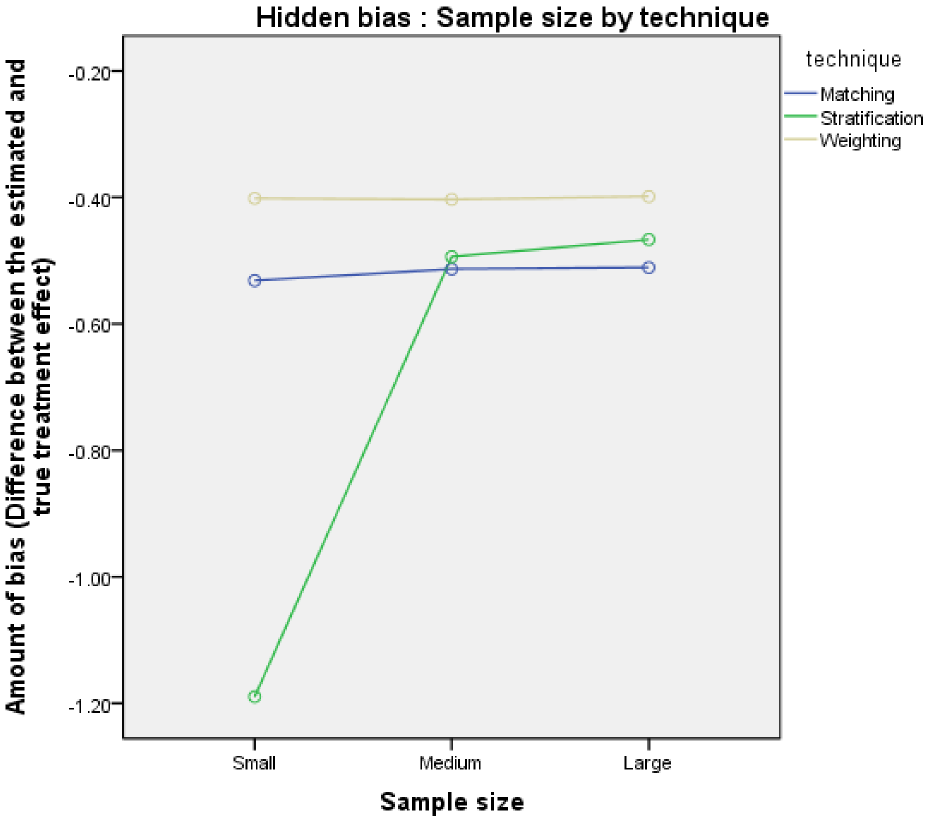

Figure 8

Interaction between Propensity Score Technique and Sample Size in Hidden Bias

Simple Effect Analysis on Technique by Sample Interaction

The propensity score technique was found to have an impact under the small sample size condition, η^2 =.14 (Table 18). On the other hand, the propensity score technique had a negligible effect for the medium and large sample size conditions. Results showed that the three propensity score techniques performed differently under the small sample size condition (Table 18). The difference between the three propensity score techniques for the small sample size condition was explored using pairwise comparisons. The pairwise comparisons were performed to address the influence of sample size between propensity score techniques.

Table 18

Summary of Simple Effect Analysis on Technique by Sample Size Interaction

Source	**SS**	**df**	***F***	***p-value***	**Effect size (**η^2)
Small sample	2142.10	2	4463.61	<.001	.14
Medium sample	41.40	2	86.37	<.001	.003
Large sample	38.30	2	79.86	<.001	.004
Within	12954.90	53991			

Pairwise Comparisons as a Follow Up to the Simple Effects Analysis

There was a difference in the amount of bias in estimating the treatment effect between stratification using propensity scores and propensity score weighting techniques (η^2 =.13) (Table 19). On the other hand, there were no differences between propensity score matching and propensity score weighting and between propensity score matching and stratification using propensity scores in estimating the treatment effect under the small sample size condition (Table 19). The amount of bias in estimating the treatment effect was lower for stratification using propensity scores (M = -1.19, SD = 1.29) compared to propensity score weighting (M =-.70, SD = .28) (Table 17). The stratification approach found to have most underestimated treatment effect compared to matching and weighting under the small sample size condition.

Table 19

Summary of Pairwise Comparison Analysis of Propensity Score Technique on Small Sample

Source	**SS**	**df**	***F***	***p-value***	**Effect size (η^2)**
Matching-Stratification	5415.56	1	5415.56	<.001	.09
Matching-Weighting	210.97	1	210.97	<.001	.004
Stratification-Weighting	7764.31	1	7764.31	<.001	.13
Within	12954.92	53991			

Summary of Hidden Bias Findings

In the context of hidden bias, all three propensity score analysis techniques underestimated the treatment effect. The treatment effect estimates after correcting for selection biases were lower than the true treatment effect. The three propensity score techniques were found to perform differently in the small sample size condition. In the small sample size condition, the stratification technique was found to underestimate the treatment effect more than the matching and weighting approaches. The treatment effect estimate was closer to the true treatment effect when using the propensity score weighting adjustment approach. In addition, the presence of hidden bias found to have no impact on the performance propensity score techniques. The three propensity score techniques worked similarly despite the various levels of hidden bias.

Overall Summary from both Overt and Hidden Bias Findings

The findings for overt and hidden biases were similar. Results showed that propensity score techniques performed differently in the small sample size condition under both overt and hidden biases. There were no differences in the performance of three propensity score techniques for medium and large sample sizes for either overt or hidden biases. The level of overt bias was found to have a substantial effect on the estimation of the treatment effect. On the other hand, the level of hidden bias had no impact on estimating the treatment effect. The propensity score technique worked the best, especially at the small level of overt bias conditions. Findings suggest no difference between the three propensity score analysis techniques in multiple groups. However, practitioners need to be cautious about the use of stratification on propensity scores in small samples. The stratification on propensity score technique under the small sample size condition had the tendency to more severely underestimate the treatment effect. As a result, the treatment effect estimate is reported to be lower than the true effect. In the small sample size condition, a propensity score weighting adjustment provided a treatment effect that was closer to the true treatment effect. Results also inform the use of propensity scores in adjusting observable selection bias compared to hidden bias.

Chapter Four: Discussion

Introduction

This chapter includes a summary of the primary findings with regard to the research questions. The findings were synthesized to provide recommendations for the users of propensity score analysis techniques in multiple treatment groups. Further, the limitations of this study are presented with suggestions for further research.

Propensity score analysis techniques have received a great deal of attention from researchers working with quasi-experimental or observational data. As a result, propensity score analyses are widely used in numerous disciplines such as medicine (Austin, 2011, D’Agostino, 1998), economics (Hirano & Imben, 2008), and statistics (Rosenbaum, 2002; Stuart, 2010). The increasing use of propensity score techniques in the field is due in part to their credibility in reducing the impact of treatment-selection bias in the estimation of treatment effect using quasi-experimental or observational data. While propensity score analysis techniques serve as a tool for controlling selection bias, there remain concerns about their application.

Austin (2008), in a systematic review, raised concerns about the application of propensity score analysis techniques among researchers. Researchers were found to have misused or misunderstood the application of propensity score analysis (Stuart, 2008). As a consequence of poor application of propensity score analysis, researchers such as Caliendo and Koeping (2008) and D'Agostino (1998) started creating guidelines suggesting best practices for the use of propensity score analysis. Many of the guidelines are specific to a discipline such as educational research (Randolph, Fable, Manuel, & Balloun, 2014), community psychology (Lanza, Moore & Butera, 2013), and biostatistics (Austin, 2008; D'Agostino, 1998). These guidelines explain the practical concerns associated with every step in implementing a propensity score analysis. However, drawbacks of these guidelines are that they are designed for a specific type of propensity score analysis. For instance, the Randolph et al. (2014) paper focused on explaining the steps in the application of propensity score matching. But there are no clear guidelines available for the selection of a propensity score technique. Similarly, there is lack of guidance for selecting a propensity score technique with multiple treatment groups. Given that propensity score analysis in multiple treatment groups is new, this paper proposes some initial guidelines in the selection of a suitable propensity score technique.

Therefore, this study was conducted to determine the performance of different propensity score techniques with multiple treatment groups under various circumstances. The aim was to determine practical recommendations in deciding on a propensity score technique. Therefore, the performance of propensity score matching, stratification, and weighting techniques in multiple treatment groups were tested under different sample size conditions and levels of overt and hidden bias. The purpose was to gather information to help select an appropriate propensity score technique. In the next section the primary findings with regard to the factors influencing the performance of the propensity score techniques are discussed.

The Primary Findings of this Study

A number of critical discoveries were made about the performance of propensity score techniques in multiple treatment groups. First, the treatment effects after the propensity score adjustment were underestimated for both overt and hidden bias conditions. Next, the simulation outlined the influence of the levels of overt and hidden bias in the performance of propensity score techniques. Finally, the propensity scores were found to behave differently under various sample sizes for both overt and hidden bias. In the following section, each of the critical discoveries is discussed in depth.

Under-estimation of treatment effect under overt and hidden bias.

Overt bias. Results showed that the treatment effect after the propensity score adjustments was lower for both the overt and hidden bias conditions. The estimated treatment effect after the propensity score adjustments were made was lower than the true estimates of the effect.   Propensity score analysis is used to balance non-equivalent groups prior to any treatment estimation (Luellen, Shadish, & Clark, 2005). Covariates that cause imbalance between the groups are modeled to compute the propensity score estimate. The distributions of the covariates are forced to be similar across the groups under the conditional propensity score estimate. As a result, the observable variability within the groups is reduced to make the groups comparable. When the groups are comparable, potential explanations for the treatment effect due to differences in the covariate distribution can be ruled out. Consequently, the observed effect can be attributed solely to the treatment. In addition, the adjustments to the within group variability might have removed the random or sampling variability within the groups. Forcing the distributions to be similar could have caused loss of sampling variability. The potential reduction of the sampling and systematic selection bias variability would have yielded a smaller treatment effect estimate. These could be potential explanations for observing an underestimated treatment effect after propensity score adjustments. However, this is just my hypothesis and it needs to be explored. The presumed claims of propensity score analysis removing the random or sampling variability requires further investigation.

Hidden bias. The estimated treatment effect after propensity score adjustments was also lower than the true effect in the presence of hidden bias. Knowing that propensity score techniques can only account for observable selection bias in the data, the lower treatment effect estimate was indeed an interesting discovery. Even under hidden bias, propensity score techniques appeared to have adjusted the variability between the groups. But, the amount of bias in the treatment effect was smaller under the hidden bias compared to the overt bias condition. The treatment effect estimate under hidden bias was found to be closer to the true value compared to the treatment effect estimate under overt bias. Like in overt bias, propensity score adjustment seems to have adjusted the random or sampling variability within the groups. But, it is suspected that no adjustments were made between the groups on the observed covariates. This is because, the treatments and control groups probably had balanced covariate distributions prior to propensity score adjustments. Therefore, enforcing propensity score technique under the balanced observed distribution would not make a difference in the distribution of the covariates and in the treatment effect estimation. This is presumably the explanation for observing smaller treatment effect bias in the hidden than in the overt bias condition. Once again, this is my hypothesis and it requires further investigation.

The underestimated treatment effect is an important finding from this study. It informs users about the implications of using propensity score techniques in addressing selection bias. The treatment effect after the propensity score adjustment is expected to be lower than the true effect. Although underestimation of the treatment effect is probably better than overestimating the effect, it still introduces downward bias in estimating the treatment effect. The underestimation bias could introduce Type 2 error concerns where a substantial treatment effect is present but fails to be observed due to the propensity score analysis adjustments. This leads to the possibility of ignoring the presence of a true treatment effect. Furthermore, there is no indication of how much lower the treatment effect estimate may be after the application of a propensity score analysis technique. It would be beneficial to know how much lower the treatment estimates are in order to adjust the final effect. For example, suppose that the treatment estimate after propensity score analysis is 0.5 lower than the true estimate. Then, we can use the value to correct for the underestimated treatment effect. But, there is no method available in detecting the range of downward bias to correct for the underestimation of the treatment effect.

Propensity score techniques under different levels of overt and hidden bias.

Overt bias. The level of overt bias had an impact on the estimate of the treatment effect. The treatment effect estimates were closer to the true value in the presence of a small level of overt bias. When propensity score techniques were used with a small level of overt bias, the technique adjusts for the bias and results in a treatment effect closer to the true value. Results showed an inverse positive relationship between the level of overt bias and the amount of bias in estimating the treatment effect. As the level of bias increased by .2 units, the treatment effect value increased by .18 units. As the level of bias increased from small (.3) to medium (.5), the difference between the estimated and true treatment effect increased from -.82 to -1.00. The same association between the level of overt bias and the treatment effect estimate was observed in the comparison between medium and high levels of overt bias. Given a high level of overt bias, the treatment effect tends to be lower than the true estimate. Therefore, adjusting for smaller overt bias will have treatment effects that are closer to the true estimate. Although the adjustment works well with a small level of overt bias, there is no one best approach between matching, stratification, and weighting in correcting for overt bias in the data. This suggests that users can use any of three techniques in data with a small level of overt bias and no difference in the treatment effect estimate is likely.

Knowing that the treatment effect is sensitive to the level of overt bias, steps should be taken to model the presence of overt bias prior to implementing a propensity score analysis. But, the question is how to assess the level of bias in a dataset. Usually, the presence of selection bias is determined through balance assessment. A common balance assessment includes assessment of the standardized mean difference, independent groups t-test, and a correlation test (Guo & Fraser, 205). In some cases, the Kolmogorov-Smirnov test is used to test balance by comparing the distribution of the variables between groups (Ali et al., 2014). However, the statistical tests and standardized mean differences between the groups indicates the difference between them on each variable but not the level of bias. A possible way to assess bias is to assess the relationship between the residuals from the outcome to the potential covariates used in estimating propensity scores. Using a regression model, the covariates could be regressed on the residuals from outcomes to determine the magnitude of the relationship. The magnitude of the relationship could be an indicator of the level of overt bias. However, the application of a regression model is a suggestion and requires further exploration.

Hidden bias. Contrary to findings for the presence of overt bias, hidden bias was found to have no impact on the estimate of the treatment effect. Under different levels of hidden bias, the adjustment using propensity score techniques did not affect the estimate of the treatment effect, meaning that the level of hidden bias does not affect the performance of propensity score techniques. The advantage of this finding is confirmation of the role of propensity score techniques in adjusting for observable differences and not otherwise. Propensity score analysis adjusts the difference between the groups using the observed variables or characteristics (Luellen, Shadish, & Clark, 2005). In hidden bias, the source of the bias is unknown and is harder to adjust for. In addition, the result is also beneficial in selecting an appropriate propensity score technique. If the propensity score technique is subjected to a level of hidden bias, then the level of hidden bias will become a criterion in selecting propensity score technique. But in reality, it is harder to determine the sources and level of hidden bias in data. Learning that hidden bias has no impact on the performance of propensity score techniques was indeed encouraging. This is because it helps to narrow the options relating to the selection of propensity score techniques. Once again, the findings helped to clarify the role of propensity score techniques in the context of selection bias.

There were no differences between the three propensity score techniques under the hidden bias condition. This suggests no differences in the use of matching, weighting, and stratification approaches under the presence of hidden bias. Therefore, even if there is hidden bias, the use of matching, weighting, or stratification does not matter in treatment effect estimation. This finding is helpful in selecting a propensity score analysis technique because it helps to rule out potential considerations in deciding on the propensity score analysis technique. The presence of hidden bias is not a factor to be considered in selecting a propensity score analysis technique.

Propensity score technique in different sample sizes. The three propensity score techniques were found to work differently under the small sample size condition. In the medium or large sample size conditions, the three techniques performed similarly. This is another key finding which will help in selecting an appropriate propensity score technique. Users need to pay attention to the type of propensity score analysis technique when the sample size is small (less or equal to 200). The weighting approach was found to work better in the small sample size condition compared to the matching and stratification approaches. In weighting, all the observations in the control and treatment groups are used for the outcomes analysis (Guo & Fraser, 2015). The ability to retain the entire sample has the advantage of maintaining statistical power (Stone & Tang, 2013). In addition to that, the outcome analysis using the propensity score weights makes the treatment effect estimate doubly robust. A doubly robust treatment effect is a result of an outcome analysis combining regression analysis on the outcome with propensity scores as the weights (Funk, Westreich, Wiesen, Sturmer, Brookhart, & Davidian, 2011). The combination of the two approaches corrects the treatment effects twice and makes it doubly robust. On the other hand, stratification was found to have the lowest treatment effect estimate in the small sample size condition. In stratification, the difference between the outcome of treatment 2 and control groups (treatment effect 2 estimate) was lower compared to the difference between the outcome of treatment 1 and control (treatment effect 1 estimate).

When investigated further, the number of observations in treatment group 2 was lower than the number of observations in treatment group 1. In this study, the ratios for observations in treatment 1 versus control and treatment 2 versus control were 1:1 and 1:2. The number of observations in treatment 2 was smaller than the number of observations in treatment 1. In this study, every observation in treatment 2 had two potential control cases. The unequal number of observations within the treatment and control groups appeared to be problematic for stratification. In stratification, the propensity scores for each treatment and control group are divided into five strata. Then, the mean of the outcomes between the treatment and control groups are compared. The stratum with no treatment or control cases is discarded from the analysis. Thus means that some observations are excluded from the analysis. As a result, a truncated sample is used to make the comparisons and to estimate the treatment effect. This could be the potential explanation of observing downward bias in the treatment effect estimates after the stratification adjustment. Thus suggests a relatively equal number of observations is needed in treatment and control groups for the stratification approach. This is contradictory to the requirement for the matching approach.

In matching, a larger number of observations in control groups is preferable as it provides better matching options for the treatment cases (Austin, 2008d). In some instances, matching also recommends the use of 1 to N control observations for each treatment case (Bryer, 2013). Caliendo and Koeping (2008) suggest the use of more than one control case helps to reduce the variances and provides a better counterfactual for the treatment cases. This highlights the importance of the number of observations within each group in selecting a propensity score technique. In conclusion, researchers need to pay close attention to the sample size and the ratio of observations within the treatment and control groups prior to selection of a propensity score technique.

General Conclusion

Findings from this study provide four key pieces of information about the application of propensity score analysis in multiple treatment groups. The first key finding is that the effect of the treatment is underestimated after imposing propensity score adjustments. This is in accord with the literature on the treatment effect estimation after propensity score adjustments. In the presence of selection bias, the effect of a treatment is presumed to be biased. When propensity score adjustment is used to control for selection bias, the technique tends to reduce the effect of the treatment (Pasta, 2000). As a result, the effect of the treatment is undervalued. This confirms that propensity score adjustments will underestimate the treatment effect estimate as found in this study.

Second, the treatment effect estimates are affected by the level of overt bias. This finding underscored the need to address selection bias in the data prior to outcome analysis (Haro et al., 2006). Furthermore, the finding provided insights that propensity score techniques successfully reduced overt bias in the data. It certainly informs the role of the propensity score in controlling for overt selection bias (Rosenbaum & Rubin, 1983). But, there has been no investigation of the association between different levels of overt bias and treatment effect estimate. Moreover, there is no theory or praxis to support the importance of different levels of overt bias when estimating the treatment effect. This makes it difficult to decide if the influence of different levels of overt bias in estimating treatment effect is an occurrence in real data or an artificial phenomenon resulting from a simulation condition.

Third, propensity score analysis does not account for hidden bias. This finding supports the existing literature on the application of propensity score techniques. Propensity score analysis cannot account for hidden bias. Propensity score analysis by definition balances unequal group using the known and observed potential covariates (Pan & Bai, 2015). The definition clearly shows that propensity score analysis cannot handle unknown or unmeasured covariates in the data (Rosenbaum & Rubin, 1983). Failure in accounting for hidden bias reminds us of the limitation of propensity score analysis in correcting for selection bias. Certainly, the result suggests greater attention in collecting all relevant information about treatment assignment is needed to avoid hidden bias problems (Haro et al., 2006).

The fourth finding is that the propensity score techniques performed differently in a small sample size condition. Stratification is not a suggested technique with small sample sizes. The finding is consistent with previous literature in which the use of stratification in small samples is defined as a disadvantage in correcting for selection bias (Stone & Tang, 2013). Besides, stratification requires balanced observations between treatment and control groups. Due to the small sample, the stratum may not have enough cases of treatment or control observations to compute the treatment effect (Stone & Tang, 2013). This result clarifies that the stratification technique is not preferable under small sample sizes. On the other hand, the literature suggested the use of propensity score matching and weighting techniques under the small sample size. Given the correct propensity score model, both matching and weighting yield correct treatment effect estimates under a small sample size condition (Pirracchio, Rigon & Chevret, 2012). Given that the findings were consistent with the literature, these four key findings provide cautionary notes to the users of propensity score analysis in multiple treatment groups.

Limitations

The application of default functions in conducting propensity score analysis is considered as a limitation to this study. In R, both Trimatch and twang packages are rich in options for conducting matching and weighting. For example, there are at least two types of matching options available in the TriMatch package. In twang, users can determine ways of assessing the covariate balance between the groups in estimating a precise propensity score. But, for the purpose of this study, only the default functions of matching and weighting were tested. This limits the ability to examine the performance of different matching and weighting function options under various bias and sample size conditions.

Another limitation of the study relates to the application of propensity score analysis. In this study, a correctly specified model was used in estimating the propensity scores. However, this is not typical in reality. Although the best practices for covariate selection are based on theoretical grounds, it is still dependent on the availability of the covariates. In this study, it was presumed that all the covariates were present and known to have impacted the selection bias. The continuous and normally distributed variables to operationalize propensity scores are also considered as a constraint to this study. In reality, the covariates to model propensity scores may be dichotomous, skewed, or have missing observations. The ideal conditions of the variables used in this study restrict the generalizability of the findings. Lastly, the Monte Carlo simulation allows the researcher to manipulate and control for the design and data to investigate the performance of statistical methods (Guo & Fraser, 2015). Despite the advantages, simulation limits the generalizability of its findings. In this study, the performance of propensity score techniques was tested on 54 data conditions. This is not an exhaustive list of options of all the possible data conditions. Therefore, the results of the study are generalizable to studies with a similar scope.

Recommendations for Future Research

Only recently have propensity score analysis techniques been applied with multiple treatment groups. As a new and emerging technique, further investigation is needed to understand the strengths and weaknesses of the technique. This study was an attempt to determine the best practices of propensity score analysis in multiple treatment groups. The results and limitations of this study identified various areas that require further exploration.

In this study, no one propensity score technique was identified as superior to another. This conclusion was derived using ideal data and propensity score model conditions. When the propensity scores are correctly defined, there is no difference between the use of matching, weighting, and stratification techniques. However, it is not always possible to know all the covariates that are causing selection bias. In such cases, the propensity score model might not be well defined and not accurate in adjusting for bias. Therefore, the impact of insufficient propensity score models in treatment effect estimation needs to be explored. Research is needed to determine the sensitivity of matching, weighting, and stratification techniques with poorly defined propensity score models. The results of such research will further inform us regarding the advantages and disadvantages of the different propensity score analysis techniques in more than two groups.

More research on the characteristics of the covariates influencing treatment assignment will be beneficial in assessing the performance of different propensity score techniques. In the current study, the information regarding the treatment assignment covariates was not fully maximized. All the covariates that were in the current study were normally distributed with complete observations. This would be an ideal condition in estimating the propensity score value. There were no differences observed between matching, stratification, and weighting under the ideal covariate conditions. But, more research is required to test the consistency of the finding across different covariate settings. So, the performance of propensity score techniques should be explored on different covariate conditions such as skewed distributions or with different proportions of missing values. The findings from such research would enhance the understanding of different propensity score techniques in multiple treatment groups.

Next, further research is required in exploring ways of assessing and conducting sensitivity analysis for hidden bias in multiple treatment groups. Rosenbaum (2002) introduced sensitivity analysis in the two group analysis. But there is no clear direction available for conducting sensitivity analysis in multiple treatment groups. Determining ways of testing for hidden bias in multiple treatment groups would be a substantial contribution to the literature. Also, there is a gap in the literature regarding assessing the performance of different algorithms in computing propensity scores. In two-group studies, complex algorithms such as the Generalized Boosted Models, and neural network techniques found to yield better propensity score estimates compared to the more commonly used logistic regression (Posner, 2008). However, the application of the different propensity score estimation techniques have not been explored for studies with multiple treatment groups.

In this study, two treatments and a control group were used to test the performance of different propensity score techniques. In both treatment and stratification, the same control groups were used for estimating treatment effect 1 and treatment effect 2. The use of the same control twice would have introduced dependency in the treatment effect estimates. In matching, repeated measures ANOVA was used as a way of acknowledging the correlated outcomes. On the other hand, there were no corrections for the dependent outcomes introduced in the stratification or weighting approaches. Since stratification for the multiple treatment groups is introduced and presented for the first time in this context, it requires further investigation. In weighting, the treatment effects were computed through a regular weighted regression analysis. There were no corrections applied in the regression analysis. The debate between the use of independent and dependent outcome analysis in two groups is ongoing and no consensus is available on the best approach for estimating the treatment effect (Austin, 2008; Stuart, 2010). Therefore, more research is required to determine the consequences of treatment effect estimates before and after correcting for dependency in multiple treatment groups.

References

An, W. (2010). Bayesian propensity score estimators. *Sociological Methodology*, 40(1), 151-189.

Austin, P. C., Grootendorst, P., & Anderson, G. M. (2007). A comparison of the ability of different propensity score models to balance measured variables between treated and untreated subjects: A Monte Carlo study. *Statistics in Medicine*, *26*, 734-753.

Austin, P. C. (2008). A critical appraisal of propensity score matching in the medical literature between 19996-2003. *Statistics in Medicine*, *27*, 2037-2049. doi: 10.1002/sim.3150

Austin, P. C. (2008d). Assessing balance in baseline covariates when using many-to-one matching on the propensity-score. *Pharmacoepidemiology and Drug Safety*, *17*, 1218–1225. doi:10.1002/ pds.1674.

Austin, P. C. (2011). An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research, 46*, 399-424. doi: 10.1080/00273171.2011.568786

Braitman, L. E., & Rosenbaum, P. R. (2002). Rare outcome, common treatments: Analytic strategies using propensity scores. *Annals of Internal Medicine*, *137*(8), 693-696.

Brookhart, M. A., Schneeweiss, S., Rothman, K. J, Glynn, R. J., Avorn, J., & Stürmer T. Variable selection for propensity score models. *American Journal of Epidemiology*, *163*(12), 1149–1156.

Bryer, J. M. (2013). TriMatch: An R package for propensity score matching of non-binary treatments. Retrieved from http://cran.r-project.org/web/packages/TriMatch/ vignettes/TriMatch.pdf

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

Cohen, J. (1992). A power primer. *Psychology Bulletin*, *112*(1), 155-159.

D'Agostino, R. B. (1998). Tutorial in biostatistics: Propensity score methods for bias reduction in the comparison of a treatment t a non-randomized control group. *Statistics in Medicine*, *17*, 2265-2281.

Funk, M. J., Westreich, D., Wiesen, C., Sturmer, T., Brookhart, M. A., & Davidian, M. (2011). Doubly robust estimation of causal effects. *American Journal of Epidemiology*, *173*(7), 761-767. doi: 10.1093/aje/kwq439.

Freedman, D. A., & Berk, R. A. (2008). Weighting regressions by propensity scores. *Evaluation Review*, 32, 392–409.

Gelman, A. (2011). Causality and statistical learning. *American Journal of Sociology*, *117*(3), 955-966.

Guo, X. S., & Fraser, M. W. (2015). *Propensity score analysis: Statistical methods and applications* (2nd ed.). Thousand Oaks, CA: Sage Publications, Inc.

Harder, V. S., Stuart, E. A., & Anthony, J. C. (2010). Propensity score techniques and the assessment of measured covariate balance to test causal associations in psychological research. *Psychological Methods*, *15*(3), 234–249. http://doi.org/10.1037/a0019623

Haro, J. S. M., Kontidimas, S., Negrin, M. A., Ratcliffe, M., Suarez, D., & Windmeijer, F. (2006). Methodological aspects in the assessment of treatment effects in observational health outcomes studies. *Applied Health Economics and Health Policy*, *5*(1), 11-25.

Hinkle, D. E., Wiersma, W., & Jurs, S. G. (2003). *Applied statistics for the behavioral sciences* (5th ed.). Boston: Houghton Mifflin.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). New York: John Wiley & Sons, Inc.

Hutchinson, S., & Olmos, A. (1998). Behavior of descriptive fit indexes in confirmatory factor analysis using ordered categorical data. *Structural Equation Modeling*, *5*(4), 344-364.

Imbens, G. W. (2000). The role of the propensity score in estimating dose-response functions. *Biometrika*, *87*, 706–710.

Lanza, S.T., Moore, J. E., & Butera, N. M. (2013). Drawing causal inferences using propensity scores: A practical guide for community psychologists. *American Journal of Community Psychology*, *52*(0), 380-392. doi:10.1007/s1064-013-9604-4.

Lazarsfeld, P. F. (1959). Problem in methodology. In Merton, R. K., Broom, L., & Cottrell, L.S. (Eds.), *Sociology today: Problems and prospects*. New York: Basic Books.

Lee, B. K., Lessler, J., & Stuart, E. A. (2010). Improving propensity score weighting using machine learning. *Statistics in Medicine*, *29*, 337-346.

Lu, B., Greevy, R., Xu, X., & Beck, C. (2011). Optimal nonbipartite matching and its statistical applications. *The American Statistician*, *65*(1), 21–30.

Luellen, J. K., Shadish, W. R., & Clark, M. H. (2005). Propensity scores: An introduction and experimental test. *Evaluation Review*, *29*(6), 530-558.

Luellen, J. K. (2007). *A comparison of propensity score estimation and adjustment methods on simulated data* (Unpublished doctoral dissertation). University of Memphis, Tennessee.

Lumley, T. (2014). Survey: Analysis of complex survey samples. R package version 3.30-3.

McCaffrey, D. F., Griffin, B. A., Almirall, D., Slaughter, M. E., Ramchand, R., & Burgette, L. F. (2013). A tutorial on propensity score estimation for multiple treatment using generalized boosted model. *Statistics in Medicine*, *32*, 3388-3414.

Mitra, R., & Reiter, J. P. (2012). A comparison of two methods of estimating propensity scores after multiple imputation. *Statistical Methods in Medical Research*, *0*(0), 1-17.

Myres, J. A., & Louis, T. A. (2012). Comparing treatments via the propensity score: stratification or modeling? *Health Services & Outcomes Research Methodology*, *12*(1), 29–43. http://doi.org/10.1007/s10742-012-0080-3

Olmos, A., & Govindasamy, P. (2015). A practical guide for using propensity score weighting in R. *Practical Assessment, Research & Evaluation*, *20*(13). Retrieved from http://pareonline.net/getvn.asp?v=20&n=13

Pan, W., & Bai, H. (2015). Propensity score analysis: Concepts and issues. In Pan, W., & Bai, H (Ed.). *Propensity score analysis: Fundamentals and developments* (pp.3-19). United States: The Guilford Press.

Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization: Algorithms and complexity*. New Jersey: Prentice-Hall, Inc.

Pasta, D. J. (2000). Using propensity scores to adjust for group differences: Examples comparing alternative surgical methods. *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference,* 261-25, Indianapolis, IN.

Paxton, P., Curran, P. J., Bollen, K. A., Kirby, J., & Chen, F. (2001). Monte Carlo Experiments: Design and implementation. *Structural Equation Modeling*, *8*(2), 287-312.

Pirracchio, R., Rigon, M. R., & Chevret, S. (2012). Evaluation of the propensity score methods for estimating marginal odds ratios in case of small sample size. *Medical Research Methodology*, *12*(70). Retrieved from http://www.biomedcentral.com/1471-2288/12/70

Randolph, J. J., Falbe, K., Manuel, A. K., & Balloun, J. L. (2014). A step by step guide to propensity score matching in R. *Practical Assessment, Research & Evaluation*, *19*(18). Retrieved from http://pareonline.net/getvn.asp?v=19&n=18

Raychaudhuri, S. (2008). Proceeding from Winter Simulation Conference: *Introduction to Monte Carlo Simulation.*

Robins, J. M., Hernan, M. A., & Brumback, B. (2000). Structural models and causal inference in epidemiology. *Epidemiology*, *11*(5), 550-560.

Rodriguez, M. D., & Llorca, J. (2004). Bias. *Journal of Epidemiology Community Health*, *58*, 635–641. doi: 10.1136/jech.2003.008466

Rosenbaum, P. R. (2002). *Observational studies*. NY: Springer

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41-55.

Rosenbaum, P. R, & Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. Journal of the American Statistical Association, 79, 516-524.

Rosenbaum, P. R., & Rubin, D. B. (1985).Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39, 33–38.

Rubin, D. B. (1997). Estimating causal effects from large data sets using propensity scores. *Annals of Internal Medicine*, *127*(8S), 757-763.

Rubin, D. B., & Thomas, N. (1996). Matching using estimated propensity scores: Relating theory to practice. *International Biometric Society*, *52*(1), 249-264.

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and quasi-experimental Designs for generalized causal inference.* Belmont, CA: Wadsworth, Cengage Learning.

Shah, B. R., Laupacis, A., Hux, J. E., & Austin, P. C. (2005). Propensity score methods gave similar results to traditional regression modeling in observational studies: A systematic review. *Journal of Clinical Epidemiology*, *58*, 550-559.

Soberay, A. D. (2015). *Comparing short-term outcomes of three problem gambling treatments: A multi-group propensity score analysis* (Unpublished doctoral dissertation). University of Denver, Colorado.

Spreeuwenberg, M. D., Bartak, A., Croon, M. A., Hangenaars, J. A., Busschbach, J. J. V., Andrea, H., Twisk, J., & Stijnen, T. (2010). The multiple propensity score as a control for bias in the comparison of more than two treatment arms: An introduction from a case study in mental health. *Medical Care*, *48*(2), 166-174.

Startks, H., Diehr, P., & Curtis, R. J. (2009). The challenge of selection bias and confounding in palliative care research. Journal of Palliative Medicine, 12(2), 181-187. DOI: 10.1089/jpm.2009.9672.

Stone, C. A., & Tang, Y. (2013). Comparing propensity score methods in balancing covariates and recovering impact in small sample educational program evaluations. *Practical Assessment, Research & Evaluation*, 18(13), 1-12. Retrieved from http://pareonline.net/getvn.asp?v=18&n=13

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25(1), 1-21.

Sturmer, T., Joshi, M., Glynn, R. J., Avorn, J., Rothman, K. J., & Schneeweiss, S. (2006). A review of the application of propensity score methods yielded increasing used, advantages in specific settings, but not substantially different estimates compared with conventional multivariable methods. *Journal of Clinical Epidemiology*, *59*(5), 437-447.

Tabachnick, B. G., & Fidell, L. S (2013). *Using multivariate statistics* (6th ed). Boston: Pearson Education, Inc.

Ullman, J. B. (1996). Structural equation modeling. In Tabachnick, B. G., & Fidell, L. S. (3rd ed.), *Multivariate Statistics* (pp.709-819). New York: HarperCollins College Publishers.

Weitzen, S., Lapane, K. L., Toledano, A. Y., Hume, A. L., & Mor, V. (2004). Principles for modeling propensity scores in medical research: A systematic literature review. *Pharmacoepidemiology and Drug Safety*, *13*, 841–853.

Wilde, E. T., & Hollister, R. (2007). How close is close enough? Evaluating propensity score matching using data from a class size reduction experiment. *Journal of Policy Analysis and Management*, *26*(3), 455-477.

Winship, C., & Morgan, S. L. (1999). The estimation of causal effects from observational data. *Annual Review Sociology*, *25*, 659-706.

Wunsch, H., Zwirble, L. W.T., & Angus, D. C. (2006). Methods to adjust for bias and confounding in critical care health services research involving observational data. *Journal of Critical Care*, *21*, 1–7.

Zanutto, E., Lu, B., & Hornik, R. (2005). Using propensity score subclassification for multiple treatment dose to evaluate a National Antidrug Media Campaign. *Journal of Educational and Behavioral Statistics*, *30*(1), 59-73.

Appendix A

Correlation Matrix for Small, Medium and Large Overt and Hidden Bias

Correlation Matrix for Small Overt Bias

	x1	x2	x3	x4	x5	x6	U	v
x1	1	0.15	0.12	0.10	0	0	**0.30**	0
x2	0.15	1	0.10	0.12	0	0	**0.30**	0
x3	0.12	0.1	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	**0.30**	**0.30**	0	0	0	0	1	0
V	0	0	0	0	0	0	0	1

Correlation Matrix for Medium Overt Bias

	x1	x2	x3	x4	x5	x6	U	v
x1	1	0.15	0.12	0.10	0	0	**0.50**	0
x2	0.15	1	0.10	0.12	0	0	**0.50**	0
x3	0.12	0.10	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	**0.50**	**0.50**	0	0	0	0	1	0
V	0	0	0	0	0	0	0	1

Correlation Matrix for Large Overt Bias

	x1	x2	x3	x4	x5	x6	U	v
x1	1	0.15	0.12	0.10	0	0	**0.70**	0
x2	0.15	1	0.10	0.12	0	0	**0.70**	0
x3	0.12	0.10	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	**0.70**	**0.70**	0	0	0	0	1	0
V	0	0	0	0	0	0	0	1

Correlation Matrix for Small Hidden Bias

	x1	x2	x3	x4	x5	x6	u	v
x1	1	0.15	0.12	0.10	0	0	0	0
x2	0.15	1	0.10	0.12	0	0	0	0
x3	0.12	0.10	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	0	0	0	0	0	0	1	**0.30**
V	0	0	0	0	0	0	**0.30**	1

Correlation Matrix for Medium Hidden Bias

	x1	x2	x3	x4	x5	x6	u	v
x1	1	0.15	0.12	0.10	0	0	0	0
x2	0.15	1	0.10	0.12	0	0	0	0
x3	0.12	0.10	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	0	0	0	0	0	0	1	**0.50**
V	0	0	0	0	0	0	**0.50**	1

Correlation Matrix for Large Hidden Bias

	x1	x2	x3	x4	x5	x6	u	v
x1	1	0.15	0.12	0.10	0	0	0	0
x2	0.15	1	0.10	0.12	0	0	0	0
x3	0.12	0.10	1	0.13	0.09	0.08	0	0
x4	0.10	0.12	0.13	1	0.13	0.11	0	0
x5	0	0	0.09	0.13	1	0.15	0	0
x6	0	0	0.08	0.11	0.15	1	0	0
U	0	0	0	0	0	0	1	**0.70**
V	0	0	0	0	0	0	**0.70**	1

Appendix B

R scripts used to simulate data, conduct propensity score analysis and outcome analysis for matching, stratification and weighting

Matching

### Condition 1

#---- Condition 1 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
                .15, 1, .10, .12, 0, 0, .30, 0,
                .12, .10, 1, .13, .09, .08, 0, 0,
                .10, .12, .13, 1, .13, .11, 0, 0,
                0, 0, .09, .13, 1, .15, 0, 0,
                0, 0, .08, .11, .15, 1, 0, 0,
                .30, .30, 0, 0, 0, 0, 1, 0,
                0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)
```

```
## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"


#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))

### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data1.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))
```

```
## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)
diffT1 <-t1$estimate-diff1
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}
# Creating and labelling the final dataset
cond      <-1
technique <-1
overtbias <-1
sample    <-1

Cond1 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond1 <- rename(Cond1, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond1, "Cond1_trimatch.csv")
```

Condition 2

#---- Condition 2 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: small sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
```

```
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
                  .15, 1, .10, .12, 0, 0, .30, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .30, .30, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
```

```
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data2.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])


##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
```

```
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-2
technique <-1
overtbias <-1
sample    <-2

Cond2 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond2 <- rename(Cond1, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond2, "Cond2_trimatch.csv")
```

Condition 3

#---- Condition 3 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: large sample, =1000)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
                 .15, 1, .10, .12, 0, 0, .30, 0,
                 .12, .10, 1, .13, .09, .08, 0, 0,
                 .10, .12, .13, 1, .13, .11, 0, 0,
                 0, 0, .09, .13, 1, .15, 0, 0,
                 0, 0, .08, .11, .15, 1, 0, 0,
                 .30, .30, 0, 0, 0, 0, 1, 0,
                 0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

#--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
```

```
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
```

```
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data3.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
```

```
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-3
technique <-1
overtbias <-1
sample    <-3

Cond3 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond3 <- rename(Cond3, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond3, "Cond3_trimatch.csv")
```

Condition 4

#---- Condition 4 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
```

```
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

 sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
             .15, 1, .10, .12, 0, 0, .50, 0,
             .12, .10, 1, .13, .09, .08, 0, 0,
             .10, .12, .13, 1, .13, .11, 0, 0,
             0, 0, .09, .13, 1, .15, 0, 0,
             0, 0, .08, .11, .15, 1, 0, 0,
             .50, .50, 0, 0, 0, 0, 1, 0,
             0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
```

```
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data1.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
```

```
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-1
technique <-1
overtbias <-1
sample    <-1

Cond1 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond1 <- rename(Cond1, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond1, "Cond1_trimatch.csv")
```

Condition 5

#---- Condition 5 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
                    .15, 1, .10, .12, 0, 0, .50, 0,
                    .12, .10, 1, .13, .09, .08, 0, 0,
                    .10, .12, .13, 1, .13, .11, 0, 0,
                    0, 0, .09, .13, 1, .15, 0, 0,
                    0, 0, .08, .11, .15, 1, 0, 0,
                    .50, .50, 0, 0, 0, 0, 1, 0,
                    0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
```

```
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
```

```
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data5.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])


##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)
```

```
diffT1 <-t1$estimate-diff1
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-5
technique <-1
overtbias <-2
sample    <-2

Cond5 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond5 <- rename(Cond5, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond5, "Cond5_trimatch.csv")
```

### Condition 6

#---- Condition 6 (a: Overt bias: Medium correlation between X7 & X8 & u = .3, b: large sample, =1000)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
```

```
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                  .15, 1, .10, .12, 0, 0, .70, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .70, .70, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

#--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
```

```
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data6.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
```

```
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-6
technique <-1
overtbias <-2
sample    <-3

Cond6 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond6 <- rename(Cond6, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond6, "Cond6_trimatch.csv")
```

Condition 7

#---- Condition 7 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                    .15, 1, .10, .12, 0, 0, .70, 0,
                    .12, .10, 1, .13, .09, .08, 0, 0,
                    .10, .12, .13, 1, .13, .11, 0, 0,
                    0, 0, .09, .13, 1, .15, 0, 0,
                    0, 0, .08, .11, .15, 1, 0, 0,
                    .70, .70, 0, 0, 0, 0, 1, 0,
                    0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
```

```
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
```

```
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data7.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
```

```
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-7
technique <-1
overtbias <-3
sample    <-1

Cond7 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond7 <- rename(Cond7, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond7, "Cond7_trimatch.csv")
```

Condition 8

#---- Condition 8 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                  .15, 1, .10, .12, 0, 0, .70, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .70, .70, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
```

```
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
```

```
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data8.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])


##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)
```

```
diffT1 <-t1$estimate-diff1
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-8
technique <-1
overtbias <-3
sample    <-2

Cond8 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond8 <- rename(Cond8, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond8, "Cond8_trimatch.csv")
```

Condition 9

#---- Condition 9 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: large sample, =1000)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
```

```
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                    .15, 1, .10, .12, 0, 0, .70, 0,
                    .12, .10, 1, .13, .09, .08, 0, 0,
                    .10, .12, .13, 1, .13, .11, 0, 0,
                    0, 0, .09, .13, 1, .15, 0, 0,
                    0, 0, .08, .11, .15, 1, 0, 0,
                    .70, .70, 0, 0, 0, 0, 1, 0,
                    0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

#--Create the probability for being in Control, treatment 1 & treatment 2
## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 t1 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 t2 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+t1+t2))
 L2 <-t1/((c1+t1+t2))
 L3 <-t2/((c1+t1+t2))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
```

```
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7


### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8
### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5
### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables
## saving the dataset in csv format
write.csv(data, "data9.csv")
## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean

## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])


##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
```

```
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate, t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
diffT2 <-t2$estimate-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-9
technique <-1
overtbias <-3
sample    <-3

Cond9 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond9 <- rename(Cond9, c(X1="difft1", X2 = "difft2"))

## saving the final data set for further analysis
write.csv(Cond9, "Cond9_trimatch.csv")
```

### Condition 10

#---- Condition 10 (a: Hidden bias: Small correlation between u & v = .3, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment erorr term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
```

```
colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
## Specify the number of rows
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)

#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data1.csv")

 ##-------Phase 3: Propensity score analysis using matching
 #---Running propensity score matching using TriMatch
 ### (Caliper size = Default (.25), 1:1 match)
 library("TriMatch")
 ## Estimate propensity score and find matching triplets
 form <- ~ newx1 + newx2 + newx3 + newx4
 tpsa <- trips(data, data$gs, formu = form)
 tmatch <-trimatch(tpsa, status = FALSE)

 ## Estimating the outcome
 tmatch.out <- merge(tmatch, data$y)
 outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
 tmatch.out$id <- 1:nrow(tmatch.out)
 out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
 names(out) <- c("ID", "gment", "Outcome")
 matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

 ## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
      t2$estimate)
 ## (True mean difference is the mean difference from the original data)

 diffT1 <-t1$estimate-diff1
 diffT2 <-t2$estimate-diff2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-10
technique <-1
hiddenbias <-1
sample    <-1

Cond10 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond10 <- rename(Cond10, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond10, "Cond10_trimatch.csv")
```

Condition 11

#---- Condition 11 (a: Hidden bias: Small correlation between u & v = .3, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment erorr term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
```

```
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
             ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
             ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
             ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data1.csv")

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
(t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
(t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
    t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
```

```
 diffT2 <-t2$estimate-diff2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-11
technique <-1
hiddenbias <-1
sample    <-2


Cond11 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond11 <- rename(Cond10, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond11, "Cond11_trimatch.csv")
```

Condition 12

```
#---- Condition 12 (a: Hidden bias: Small correlation between u & v = .3, b: large sample, n=1000)

## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment erorr term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
```

```
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data1.csv")

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
(t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
(t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
     t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
```

```
 diffT2 <-t2$estimate-diff2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-12
technique <-1
hiddenbias <-1
sample    <-3


Cond12 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond12 <- rename(Cond12, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond12, "Cond12_trimatch.csv")
```

### Condition 13

#---- Condition 13 (a: Hidden bias: Medium correlation between u & v = .5, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment erorr term (u) and outcome error term (v) was set to establish
hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
               .15, 1, .10, .12, 0, 0, 0, 0,
               .12, .10, 1, .13, .09, .08, 0, 0,
               .10, .12, .13, 1, .13, .11, 0, 0,
               0, 0, .09, .13, 1, .15, 0, 0,
               0, 0, .08, .11, .15, 1, 0, 0,
               0, 0, 0, 0, 0, 0, 1, 0.5,
               0, 0, 0, 0, 0, 0, 0.5, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
```

```
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data1.csv")

 ##-------Phase 3: Propensity score analysis using matching
 #---Running propensity score matching using TriMatch
 ### (Caliper size = Default (.25), 1:1 match)
 library("TriMatch")
 ## Estimate propensity score and find matching triplets
 form <- ~ newx1 + newx2 + newx3 + newx4
 tpsa <- trips(data, data$gs, formu = form)
 tmatch <-trimatch(tpsa, status = FALSE)

 ## Estimating the outcome
 tmatch.out <- merge(tmatch, data$y)
 outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
 tmatch.out$id <- 1:nrow(tmatch.out)
 out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
 names(out) <- c("ID", "gment", "Outcome")
 matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

 ## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
    t2$estimate)
 ## (True mean difference is the mean difference from the original data)

 diffT1 <-t1$estimate-diff1
```

```
  diffT2 <-t2$estimate-diff2

  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
  }

# Creating and labelling the final dataset
cond      <-13
technique <-1
hiddenbias <-2
sample    <-1

Cond13 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond13 <- rename(Cond13, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond13, "Cond13_trimatch.csv")
```

Condition 14

#---- Condition 14(a: Hidden bias: Small correlation between u & v = .3, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment error term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                 .15, 1, .10, .12, 0, 0, 0, 0,
                 .12, .10, 1, .13, .09, .08, 0, 0,
                 .10, .12, .13, 1, .13, .11, 0, 0,
                 0, 0, .09, .13, 1, .15, 0, 0,
                 0, 0, .08, .11, .15, 1, 0, 0,
                 0, 0, 0, 0, 0, 0, 1, 0.3,
                 0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
```

```
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
           ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
           ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
           ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data1.csv")

 ##-------Phase 3: Propensity score analysis using matching
 #---Running propensity score matching using TriMatch
 ### (Caliper size = Default (.25), 1:1 match)
 library("TriMatch")
 ## Estimate propensity score and find matching triplets
 form <- ~ newx1 + newx2 + newx3 + newx4
 tpsa <- trips(data, data$gs, formu = form)
 tmatch <-trimatch(tpsa, status = FALSE)

 ## Estimating the outcome
 tmatch.out <- merge(tmatch, data$y)
 outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
 tmatch.out$id <- 1:nrow(tmatch.out)
 out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
 names(out) <- c("ID", "gment", "Outcome")
 matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

 ## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
    t2$estimate)
 ## (True mean difference is the mean difference from the original data)

 diffT1 <-t1$estimate-diff1
 diffT2 <-t2$estimate-diff2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-14
technique <-1
hiddenbias <-2
sample    <-2


Cond14 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond14 <- rename(Cond14, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond14, "Cond14_trimatch.csv")
```

Condition 15

```
#---- Condition 15 (a: Hidden bias: Medium correlation between u & v = .5, b: large sample, n=1000)

## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment erorr term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                .15, 1, .10, .12, 0, 0, 0, 0,
                .12, .10, 1, .13, .09, .08, 0, 0,
                .10, .12, .13, 1, .13, .11, 0, 0,
                0, 0, .09, .13, 1, .15, 0, 0,
                0, 0, .08, .11, .15, 1, 0, 0,
                0, 0, 0, 0, 0, 0, 1, 0.5,
                0, 0, 0, 0, 0, 0, 0.5, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
```

```
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data15.csv")

 ##-------Phase 3: Propensity score analysis using matching
 #---Running propensity score matching using TriMatch
 ### (Caliper size = Default (.25), 1:1 match)
 library("TriMatch")
 ## Estimate propensity score and find matching triplets
 form <- ~ newx1 + newx2 + newx3 + newx4
 tpsa <- trips(data, data$gs, formu = form)
 tmatch <-trimatch(tpsa, status = FALSE)

 ## Estimating the outcome
 tmatch.out <- merge(tmatch, data$y)
 outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
 tmatch.out$id <- 1:nrow(tmatch.out)
 out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
 names(out) <- c("ID", "gment", "Outcome")
 matched.size <-length(tmatch.out$Control)
 (t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
 (t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

 ## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
    t2$estimate)
 ## (True mean difference is the mean difference from the original data)

 diffT1 <-t1$estimate-diff1
 diffT2 <-t2$estimate-diff2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-15
technique <-1
hiddenbias <-2
sample    <-3


Cond15 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond15 <- rename(Cond15, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond15, "Cond15_trimatch.csv")
```

Condition 16

#---- Condition 16 (a: Hidden bias: Large correlation between u & v = .7, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment erorr term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.7,
                  0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
```

```
colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
## Specify the number of rows
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)

#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data16.csv")

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
(t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
(t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
     t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
```

```
 diffT2 <-t2$estimate-diff2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-16
technique <-1
hiddenbias <-3
sample    <-1

Cond16 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond16 <- rename(Cond16, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond16, "Cond16_trimatch.csv")
```

Condition 17

#---- Condition 17 (a: Hidden bias: Large correlation between u & v = .7, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment erorr term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.7,
                  0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
```

```
colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
## Specify the number of rows
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)

#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data17.csv")

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
(t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
(t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
    t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
```

```
 diffT2 <-t2$estimate-diff2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-7
technique <-1
hiddenbias <-3
sample    <-2


Cond17 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond17 <- rename(Cond17, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond17, "Cond17_trimatch.csv")
```

Condition 18

#---- Condition 18 (a: Hidden bias: Large correlation between u & v = .7, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Trimatch2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment erorr term (u) and outcome error term (v) was set to establish hidden bias.
reps <-1000
sum.est <-matrix (NA, nrow = reps, ncol = 63)
bal.est <-matrix (NA, nrow = reps, ncol = 26)
par.est <-matrix (NA, nrow = reps, ncol = 14)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
              .15, 1, .10, .12, 0, 0, 0, 0,
              .12, .10, 1, .13, .09, .08, 0, 0,
              .10, .12, .13, 1, .13, .11, 0, 0,
              0, 0, .09, .13, 1, .15, 0, 0,
              0, 0, .08, .11, .15, 1, 0, 0,
              0, 0, 0, 0, 0, 0, 1, 0.7,
              0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in Control, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment
  2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data18.csv")

##-------Phase 3: Propensity score analysis using matching
#---Running propensity score matching using TriMatch
### (Caliper size = Default (.25), 1:1 match)
library("TriMatch")
## Estimate propensity score and find matching triplets
form <- ~ newx1 + newx2 + newx3 + newx4
tpsa <- trips(data, data$gs, formu = form)
tmatch <-trimatch(tpsa, status = FALSE)

## Estimating the outcome
tmatch.out <- merge(tmatch, data$y)
outcomes <- grep(".out$", names(tmatch.out), perl = TRUE)
tmatch.out$id <- 1:nrow(tmatch.out)
out <- melt(tmatch.out[, c(outcomes, which(names(tmatch.out) == "id"))], id.vars = "id")
names(out) <- c("ID", "gment", "Outcome")
matched.size <-length(tmatch.out$Control)
(t1 <- t.test(x = tmatch.out$Treat1.out, y = tmatch.out$Control.out, paired = TRUE))
(t2 <- t.test(x = tmatch.out$Treat2.out, y = tmatch.out$Control.out, paired = TRUE))

## Estimating the difference between true (diff1,diff2) and estimated mean difference
(t1$estimate,
   t2$estimate)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1$estimate-diff1
```

```
  diffT2 <-t2$estimate-diff2

  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
  }

# Creating and labelling the final dataset
cond      <-18
technique <-1
hiddenbias <-3
sample    <-3


Cond18 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond18 <- rename(Cond18, c(X1="difft1", X2 = "difft2))

## saving the final data set for further analysis
write.csv(Cond18, "Cond18_trimatch.csv")
```

Stratification

### Condition 19

#---- Condition 19 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
            .15, 1, .10, .12, 0, 0, .30, 0,
            .12, .10, 1, .13, .09, .08, 0, 0,
            .10, .12, .13, 1, .13, .11, 0, 0,
            0, 0, .09, .13, 1, .15, 0, 0,
            0, 0, .08, .11, .15, 1, 0, 0,
            .30, .30, 0, 0, 0, 0, 1, 0,
            0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
```

```
colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
## Specify the number of rows
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)

#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

 ## Recoding numeric variable into string (need to rename for trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
       ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
       ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
       ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
       ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data19.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using stratification
 #---Running propensity score matching using stratification
 ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
 ## Estimate logistic regression for each model
 ### Model 1 (Treat1 vs Control)
 ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
 exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
 Model1 <- data[-exc2, ]
 ## recording the variable into dummy
 Model1$d1<-ifelse(Model1$t==1, 1,0)
 ## Checking to see if the dummy was created correctly
 freq1a <-table(Model1$d1)
 freq1a

## Step 2: Estimate the propensity score using logistic regression
```

```
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
```

```
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
```

```
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2



## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond     <-19
technique <-2
overtbias <-1
```

```
sample    <-1

Cond19 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond19 <- rename(Cond19, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond19, "Cond19_stratification.csv")
```

Condition 20

#---- Condition 20 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
           .15, 1, .10, .12, 0, 0, .30, 0,
           .12, .10, 1, .13, .09, .08, 0, 0,
           .10, .12, .13, 1, .13, .11, 0, 0,
           0, 0, .09, .13, 1, .15, 0, 0,
           0, 0, .08, .11, .15, 1, 0, 0,
           .30, .30, 0, 0, 0, 0, 1, 0,
           0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)



#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

 ## Recoding numeric variable into string (need to rename for trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data20.csv")





 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using stratification
 #---Running propensity score matching using stratification
 ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
 ## Estimate logistic regression for each model
 ### Model 1 (Treat1 vs Control)
 ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
 exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
 Model1 <- data[-exc2, ]
 ## recording the variable into dummy
 Model1$d1<-ifelse(Model1$t==1, 1,0)
 ## Checking to see if the dummy was created correctly
 freq1a <-table(Model1$d1)
```

```
  freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
 }

t2 <-ATE2(Model2)$ATE
t2


## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
```

```
cond      <-20
technique <-2
overtbias <-1
sample    <-2

Cond20 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond20 <- rename(Cond20, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond20, "Cond20_stratification.csv")
```

Condition 21

```
#---- Condition 21 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: large sample,
n=10
00)

## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
               .15, 1, .10, .12, 0, 0, .30, 0,
               .12, .10, 1, .13, .09, .08, 0, 0,
               .10, .12, .13, 1, .13, .11, 0, 0,
               0, 0, .09, .13, 1, .15, 0, 0,
               0, 0, .08, .11, .15, 1, 0, 0,
               .30, .30, 0, 0, 0, 0, 1, 0,
               0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
```

```
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

 ## Recoding numeric variable into string (need to rename for trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

#---Manipulating the variables
## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
         ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
         ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
         ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6
 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
         ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data21.csv")



 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using stratification
 #---Running propensity score matching using stratification
 ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
 ## Estimate logistic regression for each model
 ### Model 1 (Treat1 vs Control)
 ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
 exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
 Model1 <- data[-exc2, ]
 ## recording the variable into dummy
 Model1$d1<-ifelse(Model1$t==1, 1,0)
 ## Checking to see if the dummy was created correctly
 freq1a <-table(Model1$d1)
 freq1a

## Step 2: Estimate the propensity score using logistic regression
```

```
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
```

```
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
```

```
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2


## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-21
technique <-2
overtbias <-1
```

```
sample    <-3

Cond21 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond21 <- rename(Cond21, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond21, "Cond21_stratification.csv")
```

Condition 22

```
#---- Condition 22 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: small
sample, n=200)

## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
              .15, 1, .10, .12, 0, 0, .50, 0,
              .12, .10, 1, .13, .09, .08, 0, 0,
              .10, .12, .13, 1, .13, .11, 0, 0,
              0, 0, .09, .13, 1, .15, 0, 0,
              0, 0, .08, .11, .15, 1, 0, 0,
              .50, .50, 0, 0, 0, 0, 1, 0,
              0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
```

```
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)


 #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables
 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
```

```
nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data22.csv")




## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])
diff1
diff2

##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
```

```
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
```

```
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
```

```
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2
## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-22
technique <-2
overtbias <-2
sample    <-1
```

```
Cond22<-data.frame(cond, technique, overtbias, sample, par.est)
Cond22 <- rename(Cond22, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond22, "Cond22_stratification.csv")
```

Condition 23

#---- Condition 23 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
          .15, 1, .10, .12, 0, 0, .50, 0,
          .12, .10, 1, .13, .09, .08, 0, 0,
          .10, .12, .13, 1, .13, .11, 0, 0,
          0, 0, .09, .13, 1, .15, 0, 0,
          0, 0, .08, .11, .15, 1, 0, 0,
          .50, .50, 0, 0, 0, 0, 1, 0,
          0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

 ## Recoding numeric variable into string (need to rename for trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
            ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data23.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using stratification
 #---Running propensity score matching using stratification
 ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
 ## Estimate logistic regression for each model
 ### Model 1 (Treat1 vs Control)
 ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
 exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
 Model1 <- data[-exc2, ]
 ## recording the variable into dummy
 Model1$d1<-ifelse(Model1$t==1, 1,0)
 ## Checking to see if the dummy was created correctly
 freq1a <-table(Model1$d1)
 freq1a

## Step 2: Estimate the propensity score using logistic regression
```

```
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
```

```
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
```

```
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2


## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-23
technique <-2
overtbias <-2
```

```
sample    <-2

Cond23 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond23<- rename(Cond23, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond23, "Cond23_stratification.csv")
```

Condition 24

**#---- Condition 24 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: large sample, n=10
00)**

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
                 .15, 1, .10, .12, 0, 0, .50, 0,
                 .12, .10, 1, .13, .09, .08, 0, 0,
                 .10, .12, .13, 1, .13, .11, 0, 0,
                 0, 0, .09, .13, 1, .15, 0, 0,
                 0, 0, .08, .11, .15, 1, 0, 0,
                 .50, .50, 0, 0, 0, 0, 1, 0,
                 0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
```

```
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)


#--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

#---Manipulating the variables
## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
              ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
              ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
              ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)

  ### For x4, the mean difference between treatment 1 and O is set to be 0.5
  ### For x4, the mean difference between treatment 2 and o is set to be 0.6
  ### Setting the difference
  nx4 <-ifelse(data$t==1, 0.5,
              ifelse(data$t==2, 0.6, 0))
  ### saving the adjusted variable into dataset
  data$newx4<-(data$x4 + nx4)

  ##-------Phase 2: Descriptive statistics
  #---To examine the descriptive statistics of the variables

  ## saving the dataset in csv format
  write.csv(data, "data21.csv")




  ## Estimating the mean of outcome (y) in each group
  mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
  mean
  ## Estimating the mean difference
  ### diff1 (Treatment 1 vs Control)
  diff1<- (mean$x[1] - mean$x[3])
  ### diff1 (Treatment 2 vs Control)
  diff2 <-(mean$x[2] - mean$x[3])
  diff1
  diff2

  ##-------Phase 3: Propensity score analysis using stratification
  #---Running propensity score matching using stratification
  ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
  ## Estimate logistic regression for each model
  ### Model 1 (Treat1 vs Control)
  ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
  exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
  Model1 <- data[-exc2, ]
  ## recording the variable into dummy
  Model1$d1<-ifelse(Model1$t==1, 1,0)
  ## Checking to see if the dummy was created correctly
  freq1a <-table(Model1$d1)
  freq1a
```

```
## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}
```

```
## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
```

```
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2


## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond     <-24
```

```
technique <-2
overtbias <-2
sample    <-3

Cond24 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond24 <- rename(Cond24, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond24, "Cond24_stratification.csv")
```

Condition 25

#---- Condition 19 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
            .15, 1, .10, .12, 0, 0, .70, 0,
            .12, .10, 1, .13, .09, .08, 0, 0,
            .10, .12, .13, 1, .13, .11, 0, 0,
            0, 0, .09, .13, 1, .15, 0, 0,
            0, 0, .08, .11, .15, 1, 0, 0,
            .70, .70, 0, 0, 0, 0, 1, 0,
            0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
```

```
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)



#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)

  ### For x4, the mean difference between treatment 1 and O is set to be 0.5
  ### For x4, the mean difference between treatment 2 and o is set to be 0.6

  ### Setting the difference
  nx4 <-ifelse(data$t==1, 0.5,
            ifelse(data$t==2, 0.6, 0))
  ### saving the adjusted variable into dataset
  data$newx4<-(data$x4 + nx4)

  ##-------Phase 2: Descriptive statistics
  #---To examine the descriptive statistics of the variables

  ## saving the dataset in csv format
  write.csv(data, "data25.csv")



  ## Estimating the mean of outcome (y) in each group
  mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
  mean
  ## Estimating the mean difference
  ### diff1 (Treatment 1 vs Control)
  diff1<- (mean$x[1] - mean$x[3])
  ### diff1 (Treatment 2 vs Control)
  diff2 <-(mean$x[2] - mean$x[3])
  diff1
  diff2

  ##-------Phase 3: Propensity score analysis using stratification
  #---Running propensity score matching using stratification
  ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
  ## Estimate logistic regression for each model
  ### Model 1 (Treat1 vs Control)
  ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
  exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
  Model1 <- data[-exc2, ]
  ## recording the variable into dummy
  Model1$d1<-ifelse(Model1$t==1, 1,0)
  ## Checking to see if the dummy was created correctly
  freq1a <-table(Model1$d1)
  freq1a
```

```
## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}
```

```
## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
```

```
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2


## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond     <-25
```

```
technique <-2
overtbias <-3
sample    <-1

Cond25<-data.frame(cond, technique, overtbias, sample, par.est)
Cond25 <- rename(Cond25, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond25, "Cond25_stratification.csv")
```

## Condition 26

#---- Condition 26 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
             .15, 1, .10, .12, 0, 0, .70, 0,
             .12, .10, 1, .13, .09, .08, 0, 0,
             .10, .12, .13, 1, .13, .11, 0, 0,
             0, 0, .09, .13, 1, .15, 0, 0,
             0, 0, .08, .11, .15, 1, 0, 0,
             .70, .70, 0, 0, 0, 0, 1, 0,
             0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)

  ### For x4, the mean difference between treatment 1 and O is set to be 0.5
```

```
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
            ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data26.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using stratification
 #---Running propensity score matching using stratification
 ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
 ## Estimate logistic regression for each model
 ### Model 1 (Treat1 vs Control)
 ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
 exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
 Model1 <- data[-exc2, ]
 ## recording the variable into dummy
 Model1$d1<-ifelse(Model1$t==1, 1,0)
 ## Checking to see if the dummy was created correctly
 freq1a <-table(Model1$d1)
 freq1a

## Step 2: Estimate the propensity score using logistic regression
```

```
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
```

```
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
```

```
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
    n <-tapply(Model2$d2, Model2$pscat, length)
    m <-length(Model2$d2)
    mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
    var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
    final<-data.frame(n, m, mean, var)
    final<-final[complete.cases(final),]
    names(final)[names(final)=="X0"]="mean_control"
    names(final)[names(final)=="X1"]="mean_treatment"
    names(final)[names(final)=="X0.1"]="variance_control"
    names(final)[names(final)=="X1.1"]="variance_treatment"
    tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
    return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2


## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-26
technique <-2
overtbias <-3
```

```
sample    <-2

Cond26 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond26 <- rename(Cond26, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond26, "Cond26_stratification.csv")
```

Condition 27

**#---- Condition 27 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: large sample, n=10
00)**

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                  .15, 1, .10, .12, 0, 0, .70, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .70, .70, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)


 #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

#---Manipulating the variables
## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)

  ### For x4, the mean difference between treatment 1 and O is set to be 0.5
  ### For x4, the mean difference between treatment 2 and o is set to be 0.6
  ### Setting the difference
  nx4 <-ifelse(data$t==1, 0.5,
            ifelse(data$t==2, 0.6, 0))
  ### saving the adjusted variable into dataset
  data$newx4<-(data$x4 + nx4)

  ##-------Phase 2: Descriptive statistics
  #---To examine the descriptive statistics of the variables

  ## saving the dataset in csv format
  write.csv(data, "data27.csv")




  ## Estimating the mean of outcome (y) in each group
  mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
  mean
  ## Estimating the mean difference
  ### diff1 (Treatment 1 vs Control)
  diff1<- (mean$x[1] - mean$x[3])
  ### diff1 (Treatment 2 vs Control)
  diff2 <-(mean$x[2] - mean$x[3])
  diff1
  diff2

  ##-------Phase 3: Propensity score analysis using stratification
  #---Running propensity score matching using stratification
  ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
  ## Estimate logistic regression for each model
  ### Model 1 (Treat1 vs Control)
  ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
  exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
  Model1 <- data[-exc2, ]
  ## recording the variable into dummy
  Model1$d1<-ifelse(Model1$t==1, 1,0)
  ## Checking to see if the dummy was created correctly
  freq1a <-table(Model1$d1)
  freq1a
```

```
## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}
```

```
## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a

### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)

## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
```

```
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)

ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
  }

t2 <-ATE2(Model2)$ATE
t2



## Average Treatment Effect 2 (ATE2) before trimming
### Extract and saving the ATE2
diffT2 <-t2-diff2
diffT2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond     <-27
```

```
technique <-2
overtbias <-3
sample    <-3

Cond27 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond27 <- rename(Cond27, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond27, "Cond27_stratification.csv")
```

Condition 28

#---- Condition 28 (a: Hidden bias: Small correlation between u & v = .3, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

 sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
```

```
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
         ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
         ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
         ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6
 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
         ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data28.csv")


 ##-------Phase 3: Propensity score analysis using stratification
 #---Running propensity score matching using stratification
 ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
 ## Estimate logistic regression for each model
 ### Model 1 (Treat1 vs Control)
 ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
 exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
 Model1 <- data[-exc2, ]
 ## Checking to see if the cases were removed correctly
 freq1 <-table(Model1$t)
 freq1
 ## recording the variable into dummy
 Model1$d1<-ifelse(Model1$t==1, 1,0)
 ## Checking to see if the dummy was created correctly
 freq1a <-table(Model1$d1)
 freq1a

 ## Step 2: Estimate the propensity score using logistic regression
 ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
 summary(ps)
 psvalue <- predict(ps, type = "response")
 Model1<- cbind(Model1, psvalue)
 boxplot(Model1$psvalue~Model1$d1)
 ## Visualizing the density plot
 ## create value labels
 p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
 ## psvalue for treatment 1 vs 0
```

```
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)



ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)
```

```
diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))
```

```
 ## Estimating treatment effect (Treatment 2 vs 0)

 ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-28
technique <-2
hiddenbias <-1
sample    <-1

Cond28 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond28 <- rename(Cond28, c(X1="difft1", X2 = "difft2)
## saving the final data set for further analysis
write.csv(Cond28, "Cond28_stratification.csv")
```

Condition 29

#---- Condition 29 (a: Hidden bias: Small correlation between u & v = .3, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data29.csv")


##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## Checking to see if the cases were removed correctly
freq1 <-table(Model1$t)
freq1
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
```

```
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)



ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)
```

```
diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))
```

```
 ## Estimating treatment effect (Treatment 2 vs 0)

 ATE2 <-function (Model2){
  n <-tapply(Model2$d2, Model2$pscat, length)
  m <-length(Model2$d2)
  mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
  var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-29
technique <-2
hiddenbias <-1
sample    <-2

Cond29 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond29 <- rename(Cond29, c(X1="difft1", X2 = "difft2)
## saving the final data set for further analysis
write.csv(Cond29, "Cond29_stratification.csv")
```

## Condition 30

#---- Condition 30 (a: Hidden bias: Small correlation between u & v = .3, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and treatment 2 vs 0. The means between the variables were set to be different.

### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data30.csv")



##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## Checking to see if the cases were removed correctly
freq1 <-table(Model1$t)
freq1
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
```

```
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)

ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
```

```
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)
```

```
 ATE2 <-function (Model2){
  n <-tapply(Model2$d2, Model2$pscat, length)
  m <-length(Model2$d2)
  mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
  var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-30
technique <-2
hiddenbias <-1
sample    <-3

Cond30 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond30 <- rename(Cond30, c(X1="difft1", X2 = "difft2)
## saving the final data set for further analysis
write.csv(Cond30, "Cond30_stratification.csv")
```

Condition 31

#---- Condition 31 (a: Hidden bias: Medium correlation between u & v = .5, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                   .15, 1, .10, .12, 0, 0, 0, 0,
                   .12, .10, 1, .13, .09, .08, 0, 0,
                   .10, .12, .13, 1, .13, .11, 0, 0,
                   0, 0, .09, .13, 1, .15, 0, 0,
                   0, 0, .08, .11, .15, 1, 0, 0,
                   0, 0, 0, 0, 0, 0, 1, 0.3,
                   0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
         ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data31.csv")



##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## Checking to see if the cases were removed correctly
freq1 <-table(Model1$t)
freq1
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)



ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1
```

```
## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
```

```
 # Add a legend (the color numbers start from 2 and go up)
 legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

 ## Estimating treatment effect (Treatment 2 vs 0)

 ATE2 <-function (Model2){
  n <-tapply(Model2$d2, Model2$pscat, length)
  m <-length(Model2$d2)
  mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
  var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-31
technique <-2
hiddenbias <-2
sample    <-1

Cond31 <-data.frame(cond, technique, hiddenbias, sample, par.est)
```

```
Cond31 <- rename(Cond31, c(X1="difft1", X2 = "difft2)
## saving the final data set for further analysis
write.csv(Cond31, "Cond31_stratification.csv")
```

Condition 32

#---- Condition 32 (a: Hidden bias: Medium correlation between u & v = .3, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                   .15, 1, .10, .12, 0, 0, 0, 0,
                   .12, .10, 1, .13, .09, .08, 0, 0,
                   .10, .12, .13, 1, .13, .11, 0, 0,
                   0, 0, .09, .13, 1, .15, 0, 0,
                   0, 0, .08, .11, .15, 1, 0, 0,
                   0, 0, 0, 0, 0, 0, 1, 0.5,
                   0, 0, 0, 0, 0, 0, 0.5, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and treatment 2 vs 0. The means between the variables were set to be different.

### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data32.csv")


##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## Checking to see if the cases were removed correctly
freq1 <-table(Model1$t)
freq1
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
```

```
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)



ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)
```

```
diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))
```

```
 ## Estimating treatment effect (Treatment 2 vs 0)

 ATE2 <-function (Model2){
  n <-tapply(Model2$d2, Model2$pscat, length)
  m <-length(Model2$d2)
  mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
  var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-32
technique <-2
hiddenbias <-2
sample    <-2

Cond32 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond32 <- rename(Cond32, c(X1="difft1", X2 = "difft2")
## saving the final data set for further analysis
write.csv(Cond32, "Cond32_stratification.csv")
```

Condition 33

#---- Condition 33 (a: Hidden bias: Medium correlation between u & v = .5, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.5,
                  0, 0, 0, 0, 0, 0, 0.5, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and treatment 2 vs 0. The means between the variables were set to be different.

### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data33.csv")


##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## Checking to see if the cases were removed correctly
freq1 <-table(Model1$t)
freq1
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
```

```
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)



ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
```

```
 legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

 ## Estimating treatment effect (Treatment 2 vs 0)

 ATE2 <-function (Model2){
  n <-tapply(Model2$d2, Model2$pscat, length)
  m <-length(Model2$d2)
  mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
  var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-33
technique <-2
hiddenbias <-2
sample    <-3

Cond33 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond33 <- rename(Cond33, c(X1="difft1", X2 = "difft2)
## saving the final data set for further analysis
write.csv(Cond33, "Cond33_stratification.csv")
```

## Condition 34

#---- Condition 34 (a: Hidden bias: Large correlation between u & v = .7, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.7,
                  0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)



  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data34.csv")



##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## Checking to see if the cases were removed correctly
freq1 <-table(Model1$t)
freq1
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)



ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1
```

```
## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
```

```
 # Add a legend (the color numbers start from 2 and go up)
 legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

 ## Estimating treatment effect (Treatment 2 vs 0)

 ATE2 <-function (Model2){
  n <-tapply(Model2$d2, Model2$pscat, length)
  m <-length(Model2$d2)
  mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
  var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-34
technique <-2
hiddenbias <-3
sample    <-1

Cond34 <-data.frame(cond, technique, hiddenbias, sample, par.est)
```

```
Cond34 <- rename(Cond34, c(X1="difft1", X2 = "difft2)
## saving the final data set for further analysis
write.csv(Cond34, "Cond34_stratification.csv")
```

Condition 35

#---- Condition 35 (a: Hidden bias: Large correlation between u & v = .7, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.7,
                  0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6
 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data35.csv")


 ##-------Phase 3: Propensity score analysis using stratification
 #---Running propensity score matching using stratification
 ### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
 ## Estimate logistic regression for each model
 ### Model 1 (Treat1 vs Control)
 ## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
 exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
 Model1 <- data[-exc2, ]
 ## Checking to see if the cases were removed correctly
 freq1 <-table(Model1$t)
 freq1
 ## recording the variable into dummy
 Model1$d1<-ifelse(Model1$t==1, 1,0)
 ## Checking to see if the dummy was created correctly
 freq1a <-table(Model1$d1)
 freq1a

 ## Step 2: Estimate the propensity score using logistic regression
 ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
 summary(ps)
 psvalue <- predict(ps, type = "response")
 Model1<- cbind(Model1, psvalue)
 boxplot(Model1$psvalue~Model1$d1)
 ## Visualizing the density plot
 ## create value labels
 p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
 ## psvalue for treatment 1 vs 0
```

```
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)



ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)
```

```
diffT1 <-t1-diff1
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))
```

```
 ## Estimating treatment effect (Treatment 2 vs 0)

 ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-35
technique <-2
hiddenbias <-3
sample    <-2

Cond35 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond35 <- rename(Cond35, c(X1="difft1", X2 = "difft2")
## saving the final data set for further analysis
write.csv(Cond35, "Cond35_stratification.csv")
```

## Condition 36

#---- Condition 36 (a: Hidden bias: Large correlation between u & v = .7, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("C:/Users/student/Desktop/Stratification2_simulation")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.7,
                  0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and treatment 2 vs 0. The means between the variables were set to be different.

### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data36.csv")


##-------Phase 3: Propensity score analysis using stratification
#---Running propensity score matching using stratification
### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate logistic regression for each model
### Model 1 (Treat1 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 1 and control)
exc2 <- which(with(data, (data$t==2))) # exclude cases assigned under treatment 2
Model1 <- data[-exc2, ]
## Checking to see if the cases were removed correctly
freq1 <-table(Model1$t)
freq1
## recording the variable into dummy
Model1$d1<-ifelse(Model1$t==1, 1,0)
## Checking to see if the dummy was created correctly
freq1a <-table(Model1$d1)
freq1a

## Step 2: Estimate the propensity score using logistic regression
ps<- glm(d1 ~ newx1 + newx2 + newx3 + newx4, data =Model1, family = binomial())
summary(ps)
psvalue <- predict(ps, type = "response")
Model1<- cbind(Model1, psvalue)
boxplot(Model1$psvalue~Model1$d1)
## Visualizing the density plot
## create value labels
p1<- factor(Model1$d1, levels= c(0,1), labels = c("Ct", "t1"))
## psvalue for treatment 1 vs 0
```

```
sm.density.compare(Model1$psvalue, p1, xlab="psvalue")
title(main="psvalues Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p1), fill=2+(0:nlevels(p1)))

## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model1<- Model1[order(psvalue),]
quintiles <- quantile(Model1$psvalue, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model1$pscat <- cut(Model1$psvalue, breaks = quintiles, labels = 1:5, include.lowest = T)
## Estimate number of cases in each stratum
table(Model1$pscat, Model1$d1)

### a. Average Treatment Effect without trimming (Treatment 1 vs 0)
## Estimate ATE (treatment 1 vs 0)


ATE1 <-function (Model1){
  n <-tapply(Model1$d1, Model1$pscat, length)
  m <-length(Model1$d1)
  mean <- tapply(Model1$y, list(Model1$pscat, Model1$d1),mean, na.rm=TRUE)
  var <- tapply(Model1$y, list(Model1$pscat, Model1$d1),var, na.rm=TRUE)
  final<-data.frame(n, m, mean, var)
  final<-final[complete.cases(final),]
  names(final)[names(final)=="X0"]="mean_control"
  names(final)[names(final)=="X1"]="mean_treatment"
  names(final)[names(final)=="X0.1"]="variance_control"
  names(final)[names(final)=="X1.1"]="variance_treatment"
  tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
  return (setNames((list(tau)),("ATE")))
}

## Average Treatment Effect 1 (ATE1)
### Extract and saving the ATE1
t1 <-ATE1(Model1)$ATE
t1

## Estimating the difference between true (diff1) and estimated mean difference (t1)
## (True mean difference is the mean difference from the original data)

diffT1 <-t1-diff1
```

```
diffT1

##--- Stratification: ATE2 (Treatment2 vs 0)
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Model 2 (Treat2 vs Control)
## Step 1: Subsetting the data (retaining cases from treatment 2 and control)
exc1 <- which(with(data, (data$t==1))) #exclude cases assigned to treatment 1
Model2 <- data[-exc1, ]
## Checking to see if the cases were removed correctly
freq2 <-table(Model2$t)
freq2

## recording the variable into dummy
Model2$d2<-ifelse(Model2$t==2, 1,0)
## Checking to see if the dummy variable was created correctly
freq2a <-table(Model2$d2)
freq2a
### a. Average Treatment Effect without trimming (Treatment 2 vs 0)
## Estimate ATE (treatment 2 vs 0)
## Step 2: Estimate the propensity score using logistic regression
ps2<- glm(d2 ~ newx1 + newx2 + newx3 + newx4, data =Model2, family = binomial())
summary(ps2)
psvalue2 <- predict(ps2, type = "response")
Model2 <- cbind(Model2, psvalue2)
## Create equally spaced quintiles. Started by sorting the PS values in ascending order
Model2<- Model2[order(psvalue2),]
quintiles <- quantile(Model2$psvalue2, prob = seq(from = 0,to = 1, by = 0.2), na.rm = T)
Model2$pscat2 <- cut(Model2$psvalue2, breaks = quintiles, labels = 1:5, include.lowest = T)
## Checking to see the number of cases in each stratum
table(Model2$pscat2, Model2$d2)
## Visualizing the density plot
## create value labels
p2<- factor(Model2$d2, levels= c(0,1), labels = c("Ct", "t2"))
## psvalue2 for treatment 2 vs 0
sm.density.compare(Model2$psvalue2, p2, xlab="psvalue2")
title(main="psvalue2 Distribution by group")
# Add a legend (the color numbers start from 2 and go up)
legend("topright", levels(p2), fill=2+(0:nlevels(p2)))

## Estimating treatment effect (Treatment 2 vs 0)
```

```
 ATE2 <-function (Model2){
   n <-tapply(Model2$d2, Model2$pscat, length)
   m <-length(Model2$d2)
   mean <- tapply(Model2$y, list(Model2$pscat, Model2$d2),mean, na.rm=TRUE)
   var <- tapply(Model2$y, list(Model2$pscat, Model2$d2),var, na.rm=TRUE)
   final<-data.frame(n, m, mean, var)
   final<-final[complete.cases(final),]
   names(final)[names(final)=="X0"]="mean_control"
   names(final)[names(final)=="X1"]="mean_treatment"
   names(final)[names(final)=="X0.1"]="variance_control"
   names(final)[names(final)=="X1.1"]="variance_treatment"
   tau<-sum((final$n/final$m)*(final$mean_treatment-final$mean_control))
   return (setNames((list(tau)),("ATE")))
 }

 t2 <-ATE2(Model2)$ATE
 t2

 ## Average Treatment Effect 2 (ATE2) before trimming
 ### Extract and saving the ATE1
 diffT2 <-t2-diff2
 diffT2

 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
 }

# Creating and labelling the final dataset
cond      <-36
technique <-2
hiddenbias <-3
sample    <-3

Cond36 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond36 <- rename(Cond36, c(X1="difft1", X2 = "difft2")
## saving the final data set for further analysis
write.csv(Cond36, "Cond36_stratification.csv")
```

Weighting

Condition 37

#---- Condition 37 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
              .15, 1, .10, .12, 0, 0, .30, 0,
              .12, .10, 1, .13, .09, .08, 0, 0,
              .10, .12, .13, 1, .13, .11, 0, 0,
              0, 0, .09, .13, 1, .15, 0, 0,
              0, 0, .08, .11, .15, 1, 0, 0,
              .30, .30, 0, 0, 0, 0, 1, 0,
              0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)

  ### For x4, the mean difference between treatment 1 and O is set to be 0.5
```

```
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data37.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
twangdata37 <-read.csv("data37.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata37, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
## Estimating treatment effect
## Assigning weights into the dataset
twangdata37$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata37)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1

## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-37
technique <-3
overtbias <-1
sample    <-1

Cond37 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond37 <- rename(Cond37, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond37, "Cond37_twang.csv")
```

## Condition 38

#---- Condition 38 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
                  .15, 1, .10, .12, 0, 0, .30, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .30, .30, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
             ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
             ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
             ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
           ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data38.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
twangdata38 <-read.csv("data38.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata38, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
## Estimating treatment effect
## Assigning weights into the dataset
twangdata38$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata38)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1

## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-38
technique <-3
overtbias <-1
sample    <-2

Cond38 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond38 <- rename(Cond38, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond38, "Cond38_twang.csv")
```

Condition 39

#---- Condition 39 (a: Overt bias: Small correlation between X7 & X8 & u = .3, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and erorr term (u) was set to establish
overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .30, 0,
                  .15, 1, .10, .12, 0, 0, .30, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .30, .30, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)



#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data39.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
twangdata39 <-read.csv("data39.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata39, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
 ## Estimating treatment effect
## Assigning weights into the dataset
twangdata39$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata39)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1


## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-39
technique <-3
overtbias <-1
sample    <-3

Cond39 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond39 <- rename(Cond39, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond39, "Cond39_twang.csv")
```

Condition 40

#---- Condition 40 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
            .15, 1, .10, .12, 0, 0, .50, 0,
            .12, .10, 1, .13, .09, .08, 0, 0,
            .10, .12, .13, 1, .13, .11, 0, 0,
            0, 0, .09, .13, 1, .15, 0, 0,
            0, 0, .08, .11, .15, 1, 0, 0,
            .50, .50, 0, 0, 0, 0, 1, 0,
            0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
             ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data40.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata40 <-read.csv("data40.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata40, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
 ## Estimating treatment effect
## Assigning weights into the dataset
Twangdata40$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata40)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1


## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-40
technique <-3
overtbias <-2
sample    <-1

Cond40 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond40 <- rename(Cond37, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond40, "Cond40_twang.csv")
```

Condition 41

#---- Condition 41 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
              .15, 1, .10, .12, 0, 0, .50, 0,
              .12, .10, 1, .13, .09, .08, 0, 0,
              .10, .12, .13, 1, .13, .11, 0, 0,
              0, 0, .09, .13, 1, .15, 0, 0,
              0, 0, .08, .11, .15, 1, 0, 0,
              .50, .50, 0, 0, 0, 0, 1, 0,
              0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
```

```
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)



#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
             ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
             ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
             ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)

  ### For x4, the mean difference between treatment 1 and O is set to be 0.5
```

```
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data41.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata41 <-read.csv("data41.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata41, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
## Estimating treatment effect
## Assigning weights into the dataset
Twangdata41$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata41)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1


## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-41
technique <-3
overtbias <-2
sample    <-2

Cond41 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond41 <- rename(Cond41, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond41, "Cond41_twang.csv")
```

Condition 42

#---- Condition 42 (a: Overt bias: Medium correlation between X7 & X8 & u = .5, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .50, 0,
                .15, 1, .10, .12, 0, 0, .50, 0,
                .12, .10, 1, .13, .09, .08, 0, 0,
                .10, .12, .13, 1, .13, .11, 0, 0,
                0, 0, .09, .13, 1, .15, 0, 0,
                0, 0, .08, .11, .15, 1, 0, 0,
                .50, .50, 0, 0, 0, 0, 1, 0,
                0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data42.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata42 <-read.csv("data42.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata42, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
 ## Estimating treatment effect
## Assigning weights into the dataset
Twangdata42$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata42)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1


## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-42
technique <-3
overtbias <-2
sample    <-3

Cond42 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond42 <- rename(Cond42, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond42, "Cond42_twang.csv")
```

## Condition 43

#---- Condition 43 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                  .15, 1, .10, .12, 0, 0, .70, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .70, .70, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)


  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for Trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
         ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
         ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
         ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6
 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data43.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata43 <-read.csv("data43.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata43, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

## Estimating treatment effect
## Assigning weights into the dataset
Twangdata43$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata43)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1


## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-43
technique <-3
overtbias <-3
sample    <-1

Cond43 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond43 <- rename(Cond43, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond43, "Cond43_twang.csv")
```

### Condition 44

#---- Condition 44 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                  .15, 1, .10, .12, 0, 0, .70, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .70, .70, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
            ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data44.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata44 <-read.csv("data44.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata44, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
## Estimating treatment effect
## Assigning weights into the dataset
Twangdata44$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata44)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1


## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-44
technique <-3
overtbias <-3
sample    <-2

Cond44 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond44 <- rename(Cond44, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond44, "Cond44_twang.csv")
```

Condition 45

#---- Condition 45 (a: Overt bias: Large correlation between X7 & X8 & u = .7, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinnomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and erorr term (u) was set to establish overt bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 2)
for (i in 1:reps) {

sigma <- matrix(c(1, .15, .12, .10, 0, 0, .70, 0,
                  .15, 1, .10, .12, 0, 0, .70, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  .70, .70, 0, 0, 0, 0, 1, 0,
                  0, 0, 0, 0, 0, 0, 0, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)


  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for Trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
         ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
         ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
         ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6
 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 ## saving the dataset in csv format
 write.csv(data, "data45.csv")

 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata45 <-read.csv("data45.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata45, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

## Estimating treatment effect
## Assigning weights into the dataset
Twangdata45$w<-get.weights(mnps.data, stop.method = "es.mean")

design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata45)

model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
summary(model)

## Extracting the coefficients from the model
t1 <-model$coefficients [2]
t2 <-model$coefficients [3]

#--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
## (True mean difference is the mean difference from the original data)

## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
### Extract and saving the ATE1
diffT1 <-t1-diff1


## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
### Extract and saving the ATE2
diffT2 <-t2-diff2

### saving the parameters in a single dataset
par.est [i, 1] <-diffT1
par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-45
technique <-3
overtbias <-3
sample    <-3

Cond45 <-data.frame(cond, technique, overtbias, sample, par.est)
Cond45 <- rename(Cond42, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond45, "Cond45_twang.csv")
```

Condition 46

#---- Condition 46 (a: Hidden bias: Small correlation between u & v = .3, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                .15, 1, .10, .12, 0, 0, 0, 0,
                .12, .10, 1, .13, .09, .08, 0, 0,
                .10, .12, .13, 1, .13, .11, 0, 0,
                0, 0, .09, .13, 1, .15, 0, 0,
                0, 0, .08, .11, .15, 1, 0, 0,
                0, 0, 0, 0, 0, 0, 1, 0.3,
                0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)




 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
           ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
           ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
           ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data46.csv")



## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])
diff1
diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
twangdata46 <-read.csv("data46.csv")
```

```
  ## Estimate weights using propensity scores
  mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata46, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

  ## Estimating treatment effect
  ## Assigning weights into the dataset
  twangdata46$w<-get.weights(mnps.data, stop.method = "es.mean")

  design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata46)

  model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
  summary(model)

  ## Extracting the coefficients from the model
  t1 <-model$coefficients [2]
  t2 <-model$coefficients [3]

  #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
  ## (True mean difference is the mean difference from the original data)


  ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
  ### Extract and saving the ATE1
  diffT1 <-t1-diff1

  ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
  ### Extract and saving the ATE2
  diffT2 <-t2-diff2

  #---Saving information
  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-46
technique <-3
hiddenbias <-1
```

```
sample    <-1

Cond46 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond46 <- rename(Cond46, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond46, "Cond46_twang.csv")
```

Condition 47

#---- Condition 47 (a: Hidden bias: Small correlation between u & v = .3, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
```

```
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

  ## Recoding numeric variable into string (need to rename for Trimatch & twang)
  data$gs[data$t ==1] <- "Treat1"
  data$gs[data$t ==2] <- "Treat2"
  data$gs[data$t ==3] <- "Control"

  #---Manipulating the variables

  ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

  ### For x1, the mean difference between treatment 1 and O is set to be 0.8
  ### For x1, the mean difference between treatment 2 and o is set to be 0.7

  ### Setting the difference
  nx1 <-ifelse(data$t==1, 0.8,
            ifelse(data$t==2, 0.7, 0))
  ## saving the adjusted variable into dataset
  data$newx1 <-(data$x1 + nx1)

  ### For x2, the mean difference between treatment 1 and O is set to be 0.7
  ### For x2, the mean difference between treatment 2 and o is set to be 0.8

  ### Setting the difference
  nx2 <-ifelse(data$t==1, 0.7,
            ifelse(data$t==2, 0.8, 0))
  ### saving the adjusted variable into dataset
  data$newx2 <-(data$x2 + nx2)

  ### For x3, the mean difference between treatment 1 and O is set to be 0.6
  ### For x3, the mean difference between treatment 2 and o is set to be 0.5

  ### Setting the difference
  nx3 <-ifelse(data$t==1, 0.6,
            ifelse(data$t==2, 0.5, 0))
  ### saving the adjusted variable into dataset
  data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data47.csv")



## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])
diff1
diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
twangdata47 <-read.csv("data47.csv")
```

```
  ## Estimate weights using propensity scores
  mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata47, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

  ## Estimating treatment effect
  ## Assigning weights into the dataset
  twangdata47$w<-get.weights(mnps.data, stop.method = "es.mean")

  design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata47)

  model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
  summary(model)

  ## Extracting the coefficients from the model
  t1 <-model$coefficients [2]
  t2 <-model$coefficients [3]

  #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
  ## (True mean difference is the mean difference from the original data)


 ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
  ### Extract and saving the ATE1
  diffT1 <-t1-diff1

  ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
  ### Extract and saving the ATE2
  diffT2 <-t2-diff2

  #---Saving information
  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-47
technique <-3
hiddenbias <-1
```

```
sample    <-2

Cond47 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond47 <- rename(Cond47, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond47, "Cond47_twang.csv")
```

Condition 48

#---- Condition 48 (a: Hidden bias: Small correlation between u & v = .3, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.3,
                  0, 0, 0, 0, 0, 0, 0.3, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

  #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and treatment 2 vs 0. The means between the variables were set to be different.

### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data48.csv")



## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])
diff1
diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
twangdata48 <-read.csv("data48.csv")

## Estimate weights using propensity scores
```

```
  mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata48, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

  ## Estimating treatment effect
  ## Assigning weights into the dataset
  twangdata48$w<-get.weights(mnps.data, stop.method = "es.mean")

  design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata48)

  model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
  summary(model)

  ## Extracting the coefficients from the model
  t1 <-model$coefficients [2]
  t2 <-model$coefficients [3]

  #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
  ## (True mean difference is the mean difference from the original data)


## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
  ### Extract and saving the ATE1
  diffT1 <-t1-diff1

  ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
  ### Extract and saving the ATE2
  diffT2 <-t2-diff2

  #---Saving information
  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-48
technique <-3
hiddenbias <-1
sample    <-3
```

```
Cond48<-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond48<- rename(Cond48, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond48, "Cond48_twang.csv")
```

## Condition 49

#---- Condition 49 (a: Hidden bias: Medium correlation between u & v = .5, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

 sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
          .15, 1, .10, .12, 0, 0, 0, 0,
          .12, .10, 1, .13, .09, .08, 0, 0,
          .10, .12, .13, 1, .13, .11, 0, 0,
          0, 0, .09, .13, 1, .15, 0, 0,
          0, 0, .08, .11, .15, 1, 0, 0,
          0, 0, 0, 0, 0, 0, 1, 0.5,
          0, 0, 0, 0, 0, 0, 0.5, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
```

```
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for Trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6
 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data49.csv")



 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using weighting
 #---Running propensity score matching using twang
 ### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

 library("twang")     # to perform propensity score weighting
 library("survey")    # to perform weighted regression

 ## Running propensity score weighting using twang
 ### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

 ## calls the data for analysis
 twangdata49 <-read.csv("data49.csv")

 ## Estimate weights using propensity scores
```

```
  mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata49, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

  ## Estimating treatment effect
  ## Assigning weights into the dataset
  twangdata49$w<-get.weights(mnps.data, stop.method = "es.mean")

  design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata49)

  model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
  summary(model)

  ## Extracting the coefficients from the model
  t1 <-model$coefficients [2]
  t2 <-model$coefficients [3]

  #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
  ## (True mean difference is the mean difference from the original data)


  ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
  ### Extract and saving the ATE1
  diffT1 <-t1-diff1

  ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
  ### Extract and saving the ATE2
  diffT2 <-t2-diff2

  #---Saving information
  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-49
technique <-3
hiddenbias <-2
sample    <-1
```

```
Cond49 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond49 <- rename(Cond49, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond49, "Cond49_twang.csv")
```

Condition 50

#---- Condition 50 (a: Hidden bias: Medium correlation between u & v = .5, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the correlation between treatment assignment variables and error term (u) was set to establish hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.5,
                  0, 0, 0, 0, 0, 0, 0.5, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and treatment 2 vs 0. The means between the variables were set to be different.
### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
```

```
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data50.csv")


## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])
diff1
diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata50 <-read.csv("data50.csv")

## Estimate weights using propensity scores
mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata50, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
 ## Estimating treatment effect
 ## Assigning weights into the dataset
 Twangdata50$w<-get.weights(mnps.data, stop.method = "es.mean")

 design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata50)

 model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
 summary(model)

 ## Extracting the coefficients from the model
 t1 <-model$coefficients [2]
 t2 <-model$coefficients [3]

 #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
 ## (True mean difference is the mean difference from the original data)

 ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
 ### Extract and saving the ATE1
 diffT1 <-t1-diff1

 ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
 ### Extract and saving the ATE2
 diffT2 <-t2-diff2

 #---Saving information
 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-50
technique <-3
hiddenbias <-2
sample    <-2

Cond50 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond50 <- rename(Cond50, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond50, "Cond50_twang.csv")
```

Condition 51

#---- Condition 51 (a: Hidden bias: Medium correlation between u & v = .5, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

 sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
          .15, 1, .10, .12, 0, 0, 0, 0,
          .12, .10, 1, .13, .09, .08, 0, 0,
          .10, .12, .13, 1, .13, .11, 0, 0,
          0, 0, .09, .13, 1, .15, 0, 0,
          0, 0, .08, .11, .15, 1, 0, 0,
          0, 0, 0, 0, 0, 0, 1, 0.5,
          0, 0, 0, 0, 0, 0, 0.5, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)



 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)

 ## Recoding numeric variable into string (need to rename for Trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)
```

```
### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6

### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data51.csv")




## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])
diff1
diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata51 <-read.csv("data51.csv")
```

```
  ## Estimate weights using propensity scores
  mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata51, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

  ## Estimating treatment effect
  ## Assigning weights into the dataset
  Twangdata51$w<-get.weights(mnps.data, stop.method = "es.mean")

  design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata51)

  model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
  summary(model)

  ## Extracting the coefficients from the model
  t1 <-model$coefficients [2]
  t2 <-model$coefficients [3]

  #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
  ## (True mean difference is the mean difference from the original data)


  ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
  ### Extract and saving the ATE1
  diffT1 <-t1-diff1

  ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
  ### Extract and saving the ATE2
  diffT2 <-t2-diff2

  #---Saving information
  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-51
technique <-3
hiddenbias <-2
```

```
sample    <-3

Cond51<-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond51<- rename(Cond51, c(X1="difft1", X2 = "difft2"))
## saving the final data set for further analysis
write.csv(Cond51, "Cond51_twang.csv")
```

### Condition 52

#---- Condition 52 (a: Hidden bias: Large correlation between u & v = .7, b: small sample, n=200)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                  .15, 1, .10, .12, 0, 0, 0, 0,
                  .12, .10, 1, .13, .09, .08, 0, 0,
                  .10, .12, .13, 1, .13, .11, 0, 0,
                  0, 0, .09, .13, 1, .15, 0, 0,
                  0, 0, .08, .11, .15, 1, 0, 0,
                  0, 0, 0, 0, 0, 0, 1, 0.7,
                  0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=200, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
  rownames(x) <- c(1:200)
  ## saving the variables into a dataset
  data<-data.frame(x)


 #--Create the probability for being in Control, treatment 1 & treatment 2

  ## Setting the weight of the variables in estimating probabilities
  c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
  c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
  c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

  ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
  L1 <-c1/((c1+c2+c3))
  L2 <-c2/((c1+c2+c3))
  L3 <-c3/((c1+c2+c3))

  ### Estimating different probabilities of being in treatment(s) and control groups
  ### The three probabilities will be used to in a multinomial distribution to generate the groups
  data$t<-rMultinom(p=cbind(L1, L2, L3))
  ### To save the number of cases in each group
  mytable <- table(data$t)
  nt1 <-mytable[1]
  nt2 <-mytable[2]
  nct <-mytable[3]

  ## Estimate the ratio of cases in treatment(s) versus control groups
  ### Ratio of cases in treatment 1 versus control
  ratio1 <-(nt1/nct)
  ### Ratio of cases in treatment 2 versus control
  ratio2 <-(nt2/nct)

  ## Create dummy variable from categorical variable
  data$d1 <- as.numeric(data$t == 1)
  data$d2 <- as.numeric(data$t == 2)

  ## create outcome (y) variable
  data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for Trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6
 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
        ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data52.csv")



 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using weighting
 #---Running propensity score matching using twang
 ### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

 library("twang")     # to perform propensity score weighting
 library("survey")    # to perform weighted regression

 ## Running propensity score weighting using twang
 ### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

 ## calls the data for analysis
 Twangdata52 <-read.csv("data52.csv")

 ## Estimate weights using propensity scores
```

```
  mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata52, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

  ## Estimating treatment effect
  ## Assigning weights into the dataset
  Twangdata52$w<-get.weights(mnps.data, stop.method = "es.mean")

  design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata52)

  model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
  summary(model)

  ## Extracting the coefficients from the model
  t1 <-model$coefficients [2]
  t2 <-model$coefficients [3]

  #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
  ## (True mean difference is the mean difference from the original data)


  ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
  ### Extract and saving the ATE1
  diffT1 <-t1-diff1

  ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
  ### Extract and saving the ATE2
  diffT2 <-t2-diff2

  #---Saving information
  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-52
technique <-3
hiddenbias <-3
sample    <-1
```

```
Cond52 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond52 <- rename(Cond52, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond52, "Cond52_twang.csv")
```

### Condition 53

#---- Condition 53(a: Hidden bias: Large correlation between u & v = .7, b: medium sample, n=500)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

 sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
          .15, 1, .10, .12, 0, 0, 0, 0,
          .12, .10, 1, .13, .09, .08, 0, 0,
          .10, .12, .13, 1, .13, .11, 0, 0,
          0, 0, .09, .13, 1, .15, 0, 0,
          0, 0, .08, .11, .15, 1, 0, 0,
          0, 0, 0, 0, 0, 0, 1, 0.7,
          0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
 mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

 x <- rmvnorm(n=500, mean=mu, sigma=sigma, method="chol")
 ## Naming the columns in the dataset
 colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
 ## Specify the number of rows
 rownames(x) <- c(1:200)
 ## saving the variables into a dataset
 data<-data.frame(x)

 #--Create the probability for being in Control, treatment 1 & treatment 2

 ## Setting the weight of the variables in estimating probabilities
 c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
 c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
 c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

 ### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is
estimated
 L1 <-c1/((c1+c2+c3))
 L2 <-c2/((c1+c2+c3))
 L3 <-c3/((c1+c2+c3))

 ### Estimating different probabilities of being in treatment(s) and control groups
 ### The three probabilities will be used to in a multinomial distribution to generate the groups
 data$t<-rMultinom(p=cbind(L1, L2, L3))
 ### To save the number of cases in each group
 mytable <- table(data$t)
 nt1 <-mytable[1]
 nt2 <-mytable[2]
 nct <-mytable[3]

 ## Estimate the ratio of cases in treatment(s) versus control groups
 ### Ratio of cases in treatment 1 versus control
 ratio1 <-(nt1/nct)
 ### Ratio of cases in treatment 2 versus control
 ratio2 <-(nt2/nct)

 ## Create dummy variable from categorical variable
 data$d1 <- as.numeric(data$t == 1)
 data$d2 <- as.numeric(data$t == 2)

 ## create outcome (y) variable
 data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) +
(5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
 ## Recoding numeric variable into string (need to rename for Trimatch & twang)
 data$gs[data$t ==1] <- "Treat1"
 data$gs[data$t ==2] <- "Treat2"
 data$gs[data$t ==3] <- "Control"

 #---Manipulating the variables

 ## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.
 ### For x1, the mean difference between treatment 1 and O is set to be 0.8
 ### For x1, the mean difference between treatment 2 and o is set to be 0.7

 ### Setting the difference
 nx1 <-ifelse(data$t==1, 0.8,
          ifelse(data$t==2, 0.7, 0))
 ## saving the adjusted variable into dataset
 data$newx1 <-(data$x1 + nx1)

 ### For x2, the mean difference between treatment 1 and O is set to be 0.7
 ### For x2, the mean difference between treatment 2 and o is set to be 0.8

 ### Setting the difference
 nx2 <-ifelse(data$t==1, 0.7,
          ifelse(data$t==2, 0.8, 0))
 ### saving the adjusted variable into dataset
 data$newx2 <-(data$x2 + nx2)

 ### For x3, the mean difference between treatment 1 and O is set to be 0.6
 ### For x3, the mean difference between treatment 2 and o is set to be 0.5

 ### Setting the difference
 nx3 <-ifelse(data$t==1, 0.6,
          ifelse(data$t==2, 0.5, 0))
 ### saving the adjusted variable into dataset
 data$newx3 <-(data$x3 + nx3)

 ### For x4, the mean difference between treatment 1 and O is set to be 0.5
 ### For x4, the mean difference between treatment 2 and o is set to be 0.6

 ### Setting the difference
 nx4 <-ifelse(data$t==1, 0.5,
            ifelse(data$t==2, 0.6, 0))
 ### saving the adjusted variable into dataset
 data$newx4<-(data$x4 + nx4)

 ##-------Phase 2: Descriptive statistics
 #---To examine the descriptive statistics of the variables

 ## saving the dataset in csv format
 write.csv(data, "data53.csv")



 ## Estimating the mean of outcome (y) in each group
 mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
 mean
 ## Estimating the mean difference
 ### diff1 (Treatment 1 vs Control)
 diff1<- (mean$x[1] - mean$x[3])
 ### diff1 (Treatment 2 vs Control)
 diff2 <-(mean$x[2] - mean$x[3])
 diff1
 diff2

 ##-------Phase 3: Propensity score analysis using weighting
 #---Running propensity score matching using twang
 ### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

 library("twang")     # to perform propensity score weighting
 library("survey")    # to perform weighted regression

 ## Running propensity score weighting using twang
 ### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

 ## calls the data for analysis
 Twangdata53 <-read.csv("data53.csv")

 ## Estimate weights using propensity scores
 mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata53, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)
```

```
 ## Estimating treatment effect
 ## Assigning weights into the dataset
 Twangdata53$w<-get.weights(mnps.data, stop.method = "es.mean")

 design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata53)

 model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
 summary(model)

 ## Extracting the coefficients from the model
 t1 <-model$coefficients [2]
 t2 <-model$coefficients [3]

 #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
 ## (True mean difference is the mean difference from the original data)

 ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
 ### Extract and saving the ATE1
 diffT1 <-t1-diff1

 ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
 ### Extract and saving the ATE2
 diffT2 <-t2-diff2

 #---Saving information
 ### saving the parameters in a single dataset
 par.est [i, 1] <-diffT1
 par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-53
technique <-3
hiddenbias <-3
sample    <-2

Cond53 <-data.frame(cond, technique, hiddenbias, sample, par.est)
Cond53 <- rename(Cond53, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond53, "Cond53_twang.csv")
```

## Condition 54

#---- Condition 54 (a: Hidden bias: Large correlation between u & v = .7, b: large sample, n=1000)

```
## Setting the working directory and required packages
setwd("~/Desktop/R")

## Required packages
library("foreign")   # to write and save files in different format
library ("reshape")  # to rename variables
library("MASS")      # to create random numbers
library("mvtnorm")   # to create correlated random numbers
library("psych")     # to describe and estimate regression
library("miscF")     # to create random groups from multinomial distribution
library("sm")        # to create density plots

set.seed(5)
##-------Phase 1: Generating variable
#--Generate correlated variables
## Correlation matrix is specified to set the correlation between the variables. Also, the
correlation between treatment assignment variables and error term (u) was set to establish
hidden bias.
reps <-1000
par.est <-matrix (NA, nrow = reps, ncol = 4)
for (i in 1:reps) {

  sigma <- matrix(c(1, .15, .12, .10, 0, 0, 0, 0,
                 .15, 1, .10, .12, 0, 0, 0, 0,
                 .12, .10, 1, .13, .09, .08, 0, 0,
                 .10, .12, .13, 1, .13, .11, 0, 0,
                 0, 0, .09, .13, 1, .15, 0, 0,
                 0, 0, .08, .11, .15, 1, 0, 0,
                 0, 0, 0, 0, 0, 0, 1, 0.7,
                 0, 0, 0, 0, 0, 0, 0.7, 1), ncol = 8)
  mu <- c(5, 7, 9, 11, 4, 6, 0, 0)

  x <- rmvnorm(n=1000, mean=mu, sigma=sigma, method="chol")
  ## Naming the columns in the dataset
  colnames(x) <- c("x1","x2","x3","x4","x5", "x6", "u", "v")
  ## Specify the number of rows
```

```
rownames(x) <- c(1:200)
## saving the variables into a dataset
data<-data.frame(x)


#--Create the probability for being in Control, treatment 1 & treatment 2

## Setting the weight of the variables in estimating probabilities
c1 <-exp((-0.2*(data$x1)) + (-0.7*(data$x2)) + (-1.7*(data$x3)) + (-2*(data$x4)) + data$v)
c2 <-exp((-0.3*(data$x1)) + (-0.65*(data$x2)) + (-1.6*(data$x3)) + (-2.1*(data$x4)) + data$v)
c3 <-exp((-0.35*(data$x1)) + (-0.5*(data$x2)) + (-1.5*(data$x3)) + (-2.2*(data$x4)) + data$v)

### With the estimates, the probability of being in COntrol, treatment 1 and treatment 2 is estimated
L1 <-c1/((c1+c2+c3))
L2 <-c2/((c1+c2+c3))
L3 <-c3/((c1+c2+c3))

### Estimating different probabilities of being in treatment(s) and control groups
### The three probabilities will be used to in a multinomial distribution to generate the groups
data$t<-rMultinom(p=cbind(L1, L2, L3))
### To save the number of cases in each group
mytable <- table(data$t)
nt1 <-mytable[1]
nt2 <-mytable[2]
nct <-mytable[3]

## Estimate the ratio of cases in treatment(s) versus control groups
### Ratio of cases in treatment 1 versus control
ratio1 <-(nt1/nct)
### Ratio of cases in treatment 2 versus control
ratio2 <-(nt2/nct)

## Create dummy variable from categorical variable
data$d1 <- as.numeric(data$t == 1)
data$d2 <- as.numeric(data$t == 2)

## create outcome (y) variable
data$y <-((0.3*(data$x3)) + (0.4*(data$x4)) + (0.95*(data$x5)) + (0.60*(data$x6)) + (5*(data$d1)) + (15*(data$d2)) + data$u)
```

```
## Recoding numeric variable into string (need to rename for Trimatch & twang)
data$gs[data$t ==1] <- "Treat1"
data$gs[data$t ==2] <- "Treat2"
data$gs[data$t ==3] <- "Control"

#---Manipulating the variables

## Setting the treatment assignment variables to be different between treatment 1 vs 0 and
treatment 2 vs 0. The means between the variables were set to be different.

### For x1, the mean difference between treatment 1 and O is set to be 0.8
### For x1, the mean difference between treatment 2 and o is set to be 0.7

### Setting the difference
nx1 <-ifelse(data$t==1, 0.8,
        ifelse(data$t==2, 0.7, 0))
## saving the adjusted variable into dataset
data$newx1 <-(data$x1 + nx1)

### For x2, the mean difference between treatment 1 and O is set to be 0.7
### For x2, the mean difference between treatment 2 and o is set to be 0.8

### Setting the difference
nx2 <-ifelse(data$t==1, 0.7,
        ifelse(data$t==2, 0.8, 0))
### saving the adjusted variable into dataset
data$newx2 <-(data$x2 + nx2)

### For x3, the mean difference between treatment 1 and O is set to be 0.6
### For x3, the mean difference between treatment 2 and o is set to be 0.5

### Setting the difference
nx3 <-ifelse(data$t==1, 0.6,
        ifelse(data$t==2, 0.5, 0))
### saving the adjusted variable into dataset
data$newx3 <-(data$x3 + nx3)

### For x4, the mean difference between treatment 1 and O is set to be 0.5
### For x4, the mean difference between treatment 2 and o is set to be 0.6
### Setting the difference
nx4 <-ifelse(data$t==1, 0.5,
          ifelse(data$t==2, 0.6, 0))
### saving the adjusted variable into dataset
data$newx4<-(data$x4 + nx4)

##-------Phase 2: Descriptive statistics
#---To examine the descriptive statistics of the variables

## saving the dataset in csv format
write.csv(data, "data54.csv")



## Estimating the mean of outcome (y) in each group
mean <-aggregate(data$y, by=list(data$t),FUN=mean, na.rm=TRUE)
mean
## Estimating the mean difference
### diff1 (Treatment 1 vs Control)
diff1<- (mean$x[1] - mean$x[3])
### diff1 (Treatment 2 vs Control)
diff2 <-(mean$x[2] - mean$x[3])
diff1
diff2

##-------Phase 3: Propensity score analysis using weighting
#---Running propensity score matching using twang
### (number of trees =3000, stop.methods = "es. mean", "ks.mean")

library("twang")     # to perform propensity score weighting
library("survey")    # to perform weighted regression

## Running propensity score weighting using twang
### (number of trees = 3000, stop. methods = "es.mean", "ks.mean")

## calls the data for analysis
Twangdata54 <-read.csv("data54.csv")

## Estimate weights using propensity scores
```

```
  mnps.data <-mnps (gs ~ newx1 + newx2 + newx3 + newx4, data = twangdata54, estimand
="ATE", verbose = FALSE, stop.method = c("es.mean", "ks.mean"), n.trees = 3000)

  ## Estimating treatment effect
  ## Assigning weights into the dataset
  Twangdata54$w<-get.weights(mnps.data, stop.method = "es.mean")

  design.mnps <-svydesign(ids =~1, weights = ~w, data = twangdata54)

  model <-svyglm(y ~ as.factor(gs) + newx3 + newx4 + x5 + x6, design=design.mnps)
  summary(model)

  ## Extracting the coefficients from the model
  t1 <-model$coefficients [2]
  t2 <-model$coefficients [3]

  #--- Estimating the difference between true (t1, t2) and estimated mean difference (diff1, diff2)
  ## (True mean difference is the mean difference from the original data)


  ## Average Treatment Effect 1 (ATE1): Treatment 1 vs Control
  ### Extract and saving the ATE1
  diffT1 <-t1-diff1
  ## Average Treatment Effect 2 (ATE2): Treatment 2 vs Control
  ### Extract and saving the ATE2
  diffT2 <-t2-diff2

  #---Saving information
  ### saving the parameters in a single dataset
  par.est [i, 1] <-diffT1
  par.est [i, 2] <-diffT2
}

# Creating and labelling the final dataset
cond      <-54
technique <-3
hiddenbias <-3
sample    <-3

Cond54<-data.frame(cond, technique, hiddenbias, sample, par.est)
```

```
Cond54<- rename(Cond54, c(X1="difft1", X2 = "difft2))
## saving the final data set for further analysis
write.csv(Cond54, "Cond54_twang.csv")
```