P8160 - Comparing Bootstrapping Methods Report

Amy Pitts, Hun Lee, Jimmy Kelliher, Tucker Morgan, Waveley Qiu

2/11/2022

Introduction

In an observation data setting estimating the average treatment effect (ATE) can be a challenge. In an ideal world a randomized clinical trial would be conducted where participants are randomized to treatment and no treatment which in term controls for observed and unobserved confounding variables allowing the investigators to estimate the ATE. Due to the limitations around RCTs methods such as propensity-score matching have been developed to examine observational data ATE value. The propensity score is a probability associated with participant treatment assigment given the measured baseline covariates [1] . Matching similar treated participants with untreated participants based on their propensity score can help minimizes the confounding that occurs in the observed covariates. In particular this paper will be exploring nearest-neighbor propensity score matching (NNM) using the MatchIt R package [2] . It has been shown that subject who have similar propensity scores will have a similar distribution of measured baseline covariates [1] . There are suggested limitations in matching methods ability to estimate the variance of the treatment effect [3, 4] . Bootstrapping is a resampling-based approach and that draws repeated sample with replacement for the original sample to help explore statistical inference when other approaches are limited [4] . This report will explore bootstrapping methods and their ability to estimate the standard error of the estimated ATE in observational data.

Aims

The primary goal of this simulation study is to assess the performance of two bootstrap methods (detailed below) in estimating the sampling variability of treatment effects obtained from a nearest-neighbor propensity-score matching (NNM) [2] . In this study, NNM will select a treated subject at random from simulated observational data. The untreated subject with the nearest propensity score is then selected to be paired with the treated subject, without replacement. Treatment effects can then be estimated by comparing outcomes (continuous or binary) between the treated and untreated subjects. The bootstrapping methods will be used to assess the variance of estimated treatment effects.

Methods

Data Generation

The data for this simulation study were generated from a parametric model. This simulation study is interested in observational data therefore, our data generation will be modeled off of a potential observational study that could happen in the real world. For each subject, three baseline covariates (L_1, L_2, L_3) were simulated from independent standard normal distributions, $N(0, 1)$. Two of these covariates (L_1 and L_2) affected treatment selection, while two (L_2 and L_3) affected the outcome. Here the L_2 variable will be acting as the confounding since it is shared in the treatment assignment and outcome generation. The probability of treatment for each subject was determined by the following model:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \alpha_0 + \alpha_1 L_{1i} + \alpha_2 L_{2i}$$

where $\alpha_0 = \log(\frac{\pi}{1-\pi})$ as close approximation of desired treatment prevalence. Due to the nature of the logit normal there is no closed form solution to the mean of the logit normal distribution however, calculating α_0 in this fashion gives us a very close but always biased approximation.

$$\lim_{n \to \infty} \sum_{i=1}^{n} \mathcal{L}\left(\Phi^{-1}_{\mu,\sigma^2}\left(\frac{i}{n+1}\right)\right)$$

For continuous outcomes, 100 sub-populations of 100 or 1,000 subjects will be generated using the following parametric model:

$$Y_i = \beta_1 A_i + \beta_2 L_{2i} + \beta_3 L_{3i} + \epsilon_i$$

where Y_i indicates the outcome for each subject, A_i indicates the treatment status of each subject (0 or 1), L_{2i} and L_{3i} indicate observed covariate values for each subject, and ϵ_i denotes random error. Because L_{2i} affects both A_i and Y_i, it acts as a confounder in estimating the treatment effect.

For binary outcomes, the same procedure will be performed using the following parametric model:

$$\log\left(\frac{\tau_i}{1-\tau_i}\right) = \beta_0 + \beta_1 A_i + \beta_2 L_{2i} + \beta_3 L_{3i}$$

where $Y_i \sim \text{Bernoulli}(\tau_i)$. The binary outcome model does not feature an error term, as realizations of Y_i are innately subject to noise.

Here I am attempting to explain what the truth is but will need to be revised!

In the choice of parameter distributions and structure of our treatment assignment and outcome generation the true distribution can be calculated. This will allow the simulation estimates to be compared to a truth.

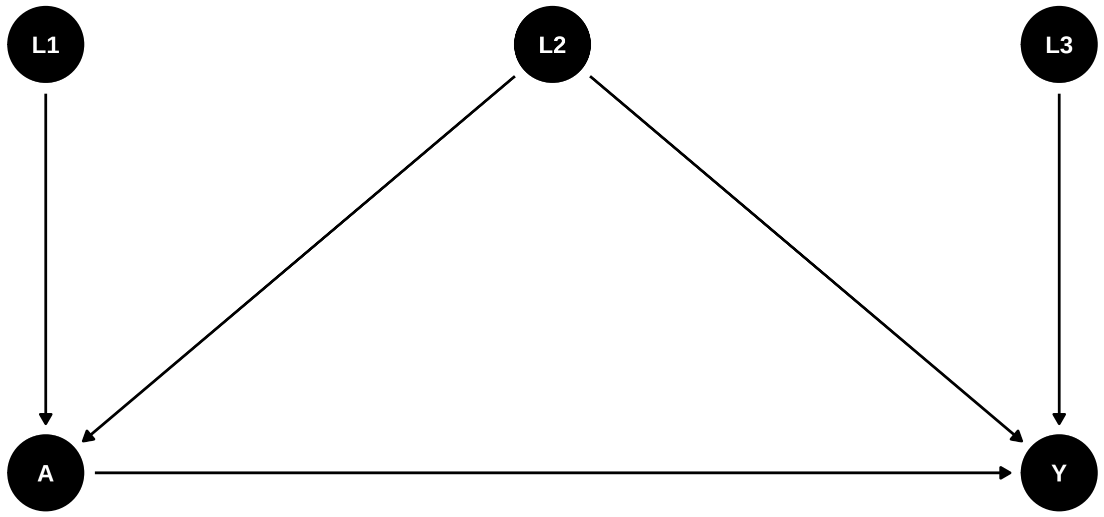
The treatment assignment for follows a logit normal with $\mu = \alpha_0$ and $\sigma^2 = \alpha_2^2 + \alpha_3^2$.

The outcome variable for the binary data follows at logit normal distribution with $\mu = \beta_1$ and $\sigma^2 = \beta_2^2 + \beta_3^2$.

The outcome variable for the continuous data data follows at normal distribution with $\mu = \beta_1$ and $\sigma^2 = \beta_2^2 + \beta_3^2$.

Thus the true treatment effect will depend on the assignment of β_1.

Figure 1. Data Generation DAG



Evaluation

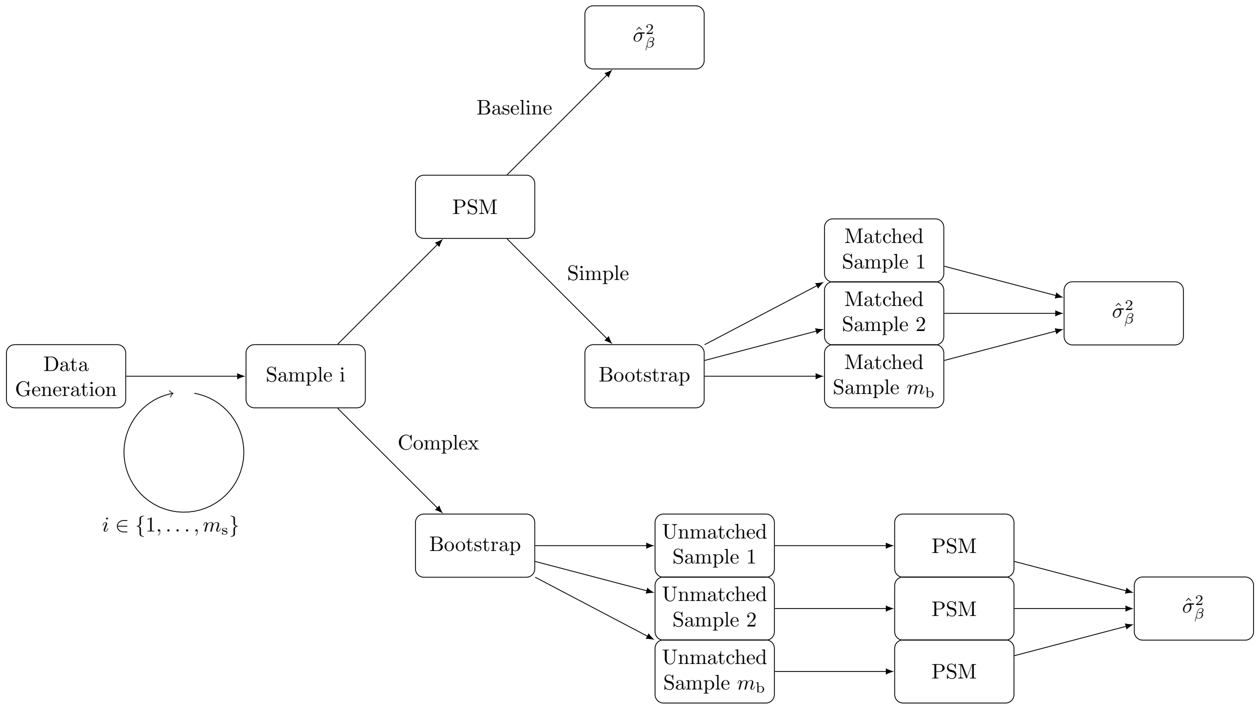
Two bootstrap methods will be assessed in this simulation: the simple bootstrap and the complex bootstrap.

In the simple bootstrap, one draws repeated samples from an original sample with replacement to imitate the process of drawing samples from a population. Here, 500 repeated samples (m_{boot}) of matched pairs ($n_{boot} = n_{sample} \cdot P(A = 1)$) will be drawn from the matched pairs of observations for each of the 100 initial samples (m_{sample}). The distribution of the estimated treatment effect ($\hat{\beta_1}$) across the 500 bootstraps is assessed for each of the 100 initial samples.

The complex bootstrap considers two additional sources of variability compared to the simple bootstrap [4] . In this approach, a sample is drawn with replacement from the original, unmatched observational data. Then by fitting a new logistic model for each bootstrap, the propensity-score model is re-estimated and re-form a propensity-score-matched sample. The treatment effect is estimated from the newly matched sample. This process is repeated 500 times (m_{boot}) for each of the 100 samples (m_{sample}).

The resulting standard error estimates, $\hat{\sigma}_\beta$, will be the primary targets of this analysis.

Figure 2. CAPTION!!



Parameters of Interest

In simulations, three parameters will vary to aid in the comparison of the two bootstrap techniques. These three parameters are dataset sample size (n_{sample}), population proportion of treated individuals (π) and the true average treatment effect (β_1). Varying the sample size and proportion of treated individuals will help make suggestions for studies that have varying amounts of participants and amount of treatment. The range the number of the sample size tried is $n_{\text{sample}} \in \{100, 1000\}$, one a low sample and one a larger sample. The range of population proportion of treated individuals $\pi \in \{0.113, 0.216, 0.313\}$. This is to approximate low, medium, and high treatment levels in a study. The true average treatment effect will take on $\beta_1 \in \{0.15, 0.30\}$ in a binary study and $\beta_1 \in \{-1, 1\}$ in a continuous study.

To compare these three parameters there are number of other parameters that will be held constant from simulation to simulations. Such as the number of datasets ($m_{\text{sample}} = 100$), the number of bootstrap re-sample ($m_{\text{boot}} = 500$), the strength of covariate correlation on treatment status (α_1, α_2), and strength of covariate correlation on outcome variable (β_2, β_3). Without loss of generality for the continuous data ($\alpha_1 = \log(1.25), \alpha_2 = \log(1.75),\ \beta_2 = 2, \beta_3 = 1$), and for the binary data ($\alpha_1 = \log(1.25), \alpha_2 = \log(1.75), \beta_2 = \log(1.75), \beta_3 = \log(1.25)$).

Performance Measures

The standard error estimates from each bootstrap method will be assessed in two ways. First, coverage rates of confidence intervals will be analyzed to assess how frequently the true average treatment effect (β_1) is included in confidence intervals using the bootstrap-estimated treatment effect ($\bar{\hat{\beta}}_1$) and estimated standard errors ($\hat{\sigma}_\beta$). Second, standard error estimates from each bootstrap method will be compared to the sample standard deviation of treatment effects of the initial samples to determine how bootstrapping aligns with a simpler approach.

Bias is also calculated using the true treatment effect. This measure helps confirm that each method is able to accurately identify the treatment effect. A 95% percent confidence interval is also constructed around the bias using the standard error.

Simulation Execution

As this simulation study contained many different components, several options were proposed to maintain program tidiness. It was ultimately determined that it would be best to divide the code into individual RMarkdown and/or RScript files along outcome type. Each of these files would define, collate, and/or execute the particular simulation scenarios of interest.

First, RScript files containing data generation functions for each outcome type were constructed. These scripts would not only generate the initial datasets of interest (according to algorithms defined previously) but also a vector drawn from a uniform distribution to provide the seeds needed to run future sampling procedures. A seed was set at the beginning of this script file in order for the results of this simulation study to be reproducible. [Appendix A]

Bootstrapping functions were then defined, in which separate procedures were written for simple bootstrapping and complex bootstrapping. Since the simple bootstrap samples from a dataset in which propensity score matched pairs had already been created, its function only needed to perform the sampling-with-replacement procedure. [Appendix B] In the function written to handle complex bootstrapping, a matching procedure was included after the sampling-with-replacement step. [Appendix C] Both of these functions were written to perform all iterations of bootstrapping required for each initial dataset (i.e., m_{boot} samples would be produced for each base sample).

Outcome functions were then constructed to produce the estimated treatment effect we were interested in. In these outcome functions, bootstrapped samples were passed through the GLM function, and we use standardization to estimate the average treatment effect of the entire study population. To this end, the pseudo-population is created where every subject gets treatment and no treatment for each sub-population and we use GLM model to obtain estimated treatment effect from bootstrapped samples. [Appendix D] Then, these estimates were summarized and collected in datasets that were indexed by scenario, outcome type, and bootstrapping method.

Finally, these functions were compiled into scripts we could pass parameters into and run through top-to-bottom to generate the results for a given scenario. To avoid manually updating the parameters that would vary, a dataset was established to describe the scenarios of interest in the study. [Appendix E] These final programs would read in the specified row of parameters and conduct the full simulation based on those values.

Some excerpts of the programming associated with this simulation study has been included in the Appendix. All programs are available in this GitHub repository.

Results

Figure 3: Binary Coverage Rates

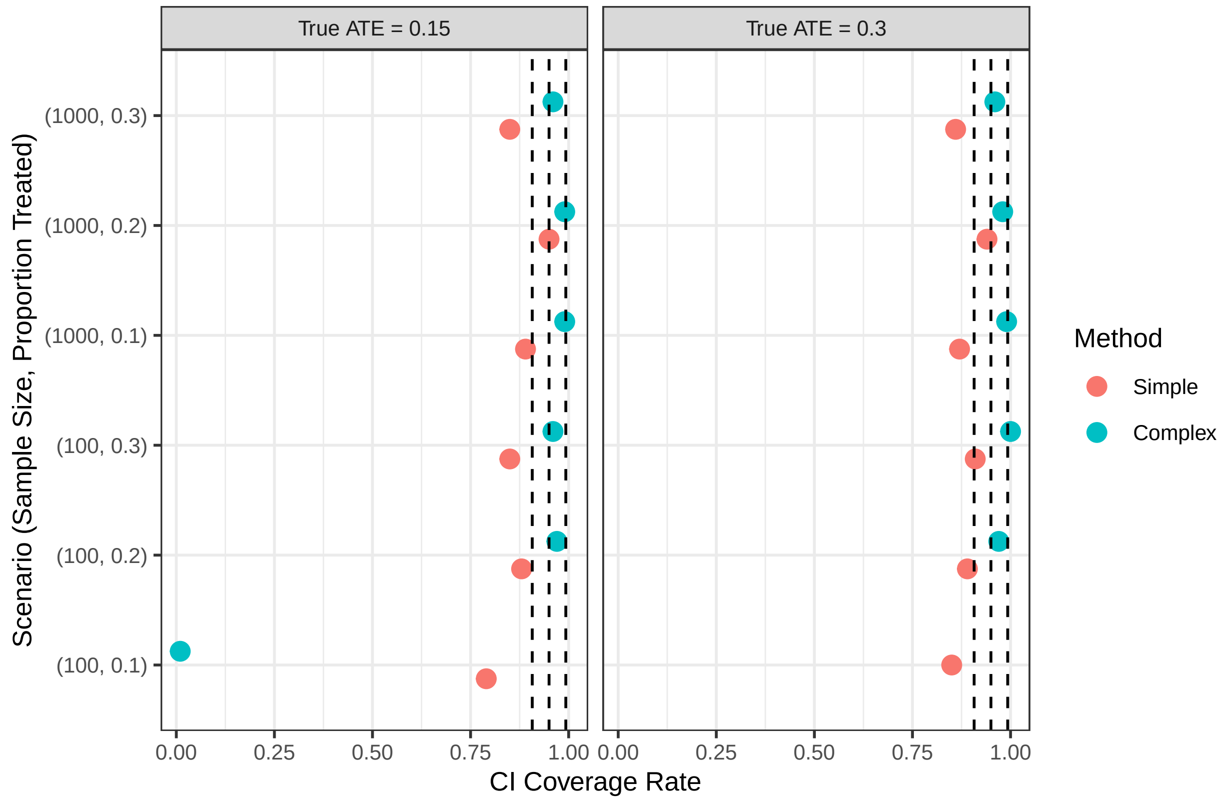
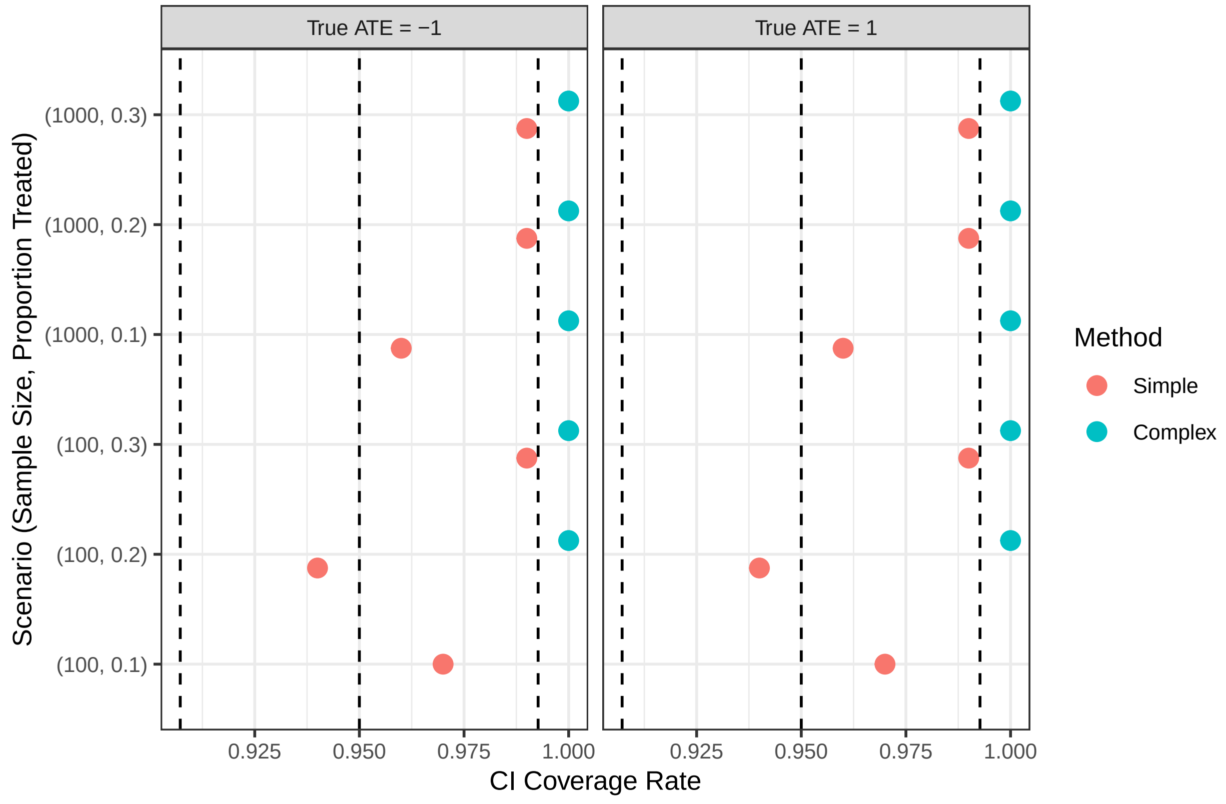


Figure 4: Continuous Coverage Rates



Analysis was performed on 24 different scenarios, 12 in a binary outcome setting and 12 in a continuous setting. To assess the standard error estimates produced by each bootstrapping method, confidence intervals were created and coverage rates calculated, see Figures 2 and 3. Each of our scenarios included 100 initial, base samples, which resulted in 100 confidence intervals. Based on the binomial distribution where n = 100, p = 0.95, a coverage rate below 90% and above 99% indicates a statistically significant under- or over-estimation of the standard error. Based on this criteria, the simple bootstrap method underestimated the standard error in five out of the six binary outcome scenarios with a lesser true average treatment effect, and the simple bootstrap underestimated the standard error of the true average treatment effect in four of the six of the scenarios involving the larger treatment effect. Conversely, the complex bootstrap method overestimated the standard error in one of the 11 scenarios. Note the complex bootstrapping method was not reliable in the scenario where $n_{sample} = 100$ and treated proportion was equal to 10%. In the scenario with a lesser treatment effect, the complex bootstrap produced a 1% coverage rate, and no estimate was produced for the scenario with a greater treatment effect.

In the continuous setting (Figure 3), the coverage rate from the simple bootstrapping method fell within the statistically significant range for all 12 scenarios. However, the complex bootstrap overestimated the standard error in 10 of the 12 scenarios. Again, the complex bootstrap was not reliable in the two scenarios where $n_{sample} = 100$ and treated proportion was equal to 10%. No estimates were produced in these instances due to sampling error wherein the bootstrapped sample contained no treated individuals, and thus no propensity score matching or treatment effect estimation could take place.

Figure 5: Binary Standard Error Estimates

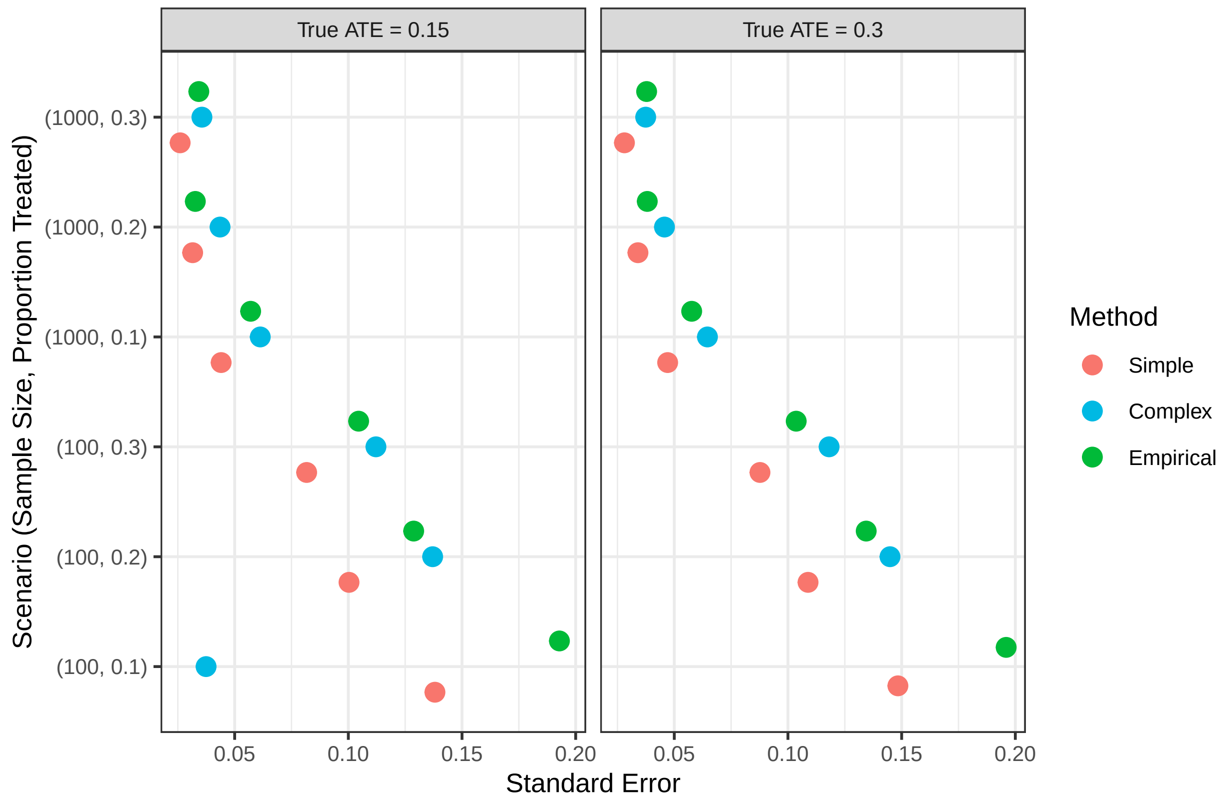
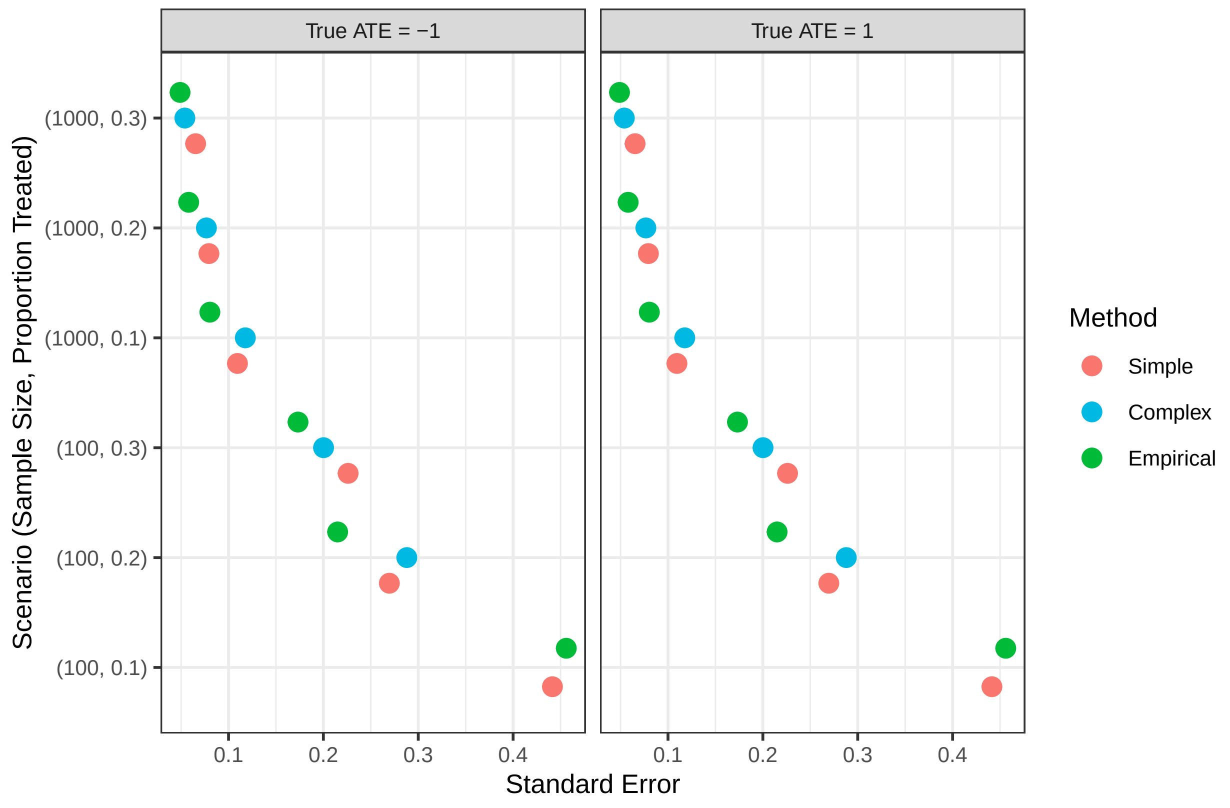


Figure 6: Continuous Standard Error Estimates



In Figure 4, the standard error estimates from the binary setting can be seen in more detail. In general, the simple bootstrap seemed to produce lower estimates of standard error compared to an empirical estimate based on the initial sample distribution before any bootstrapping, while the complex bootstrap seems to produce larger estimates of the standard error. This aligns with the observation that the complex bootstrap confidence intervals tended to have higher coverage rates compared to the simple bootstrap method. Note again the exception for complex bootstrapping for the scenario with only 100 subjects and 10% treatment proportion.

In Figure 5, a similar plot features the standard error estimates from the continuous setting. Again, the estimates from the two bootstrapping methods and the empirical measurement produce similar values. In general, the complex bootstrap produces larger estimates of standard error compared to the simple bootstrap, but the scenario in which $n_{sample} = 1000$ and treated proportion is 30% has a smaller simple bootstrap estimate compared to the complex bootstrap. These results are summarized in detail in Table 1 and Table 2. In general, standard error estimates appear to converge as sample size increases.

Figure 7: Binary Bias Distributions

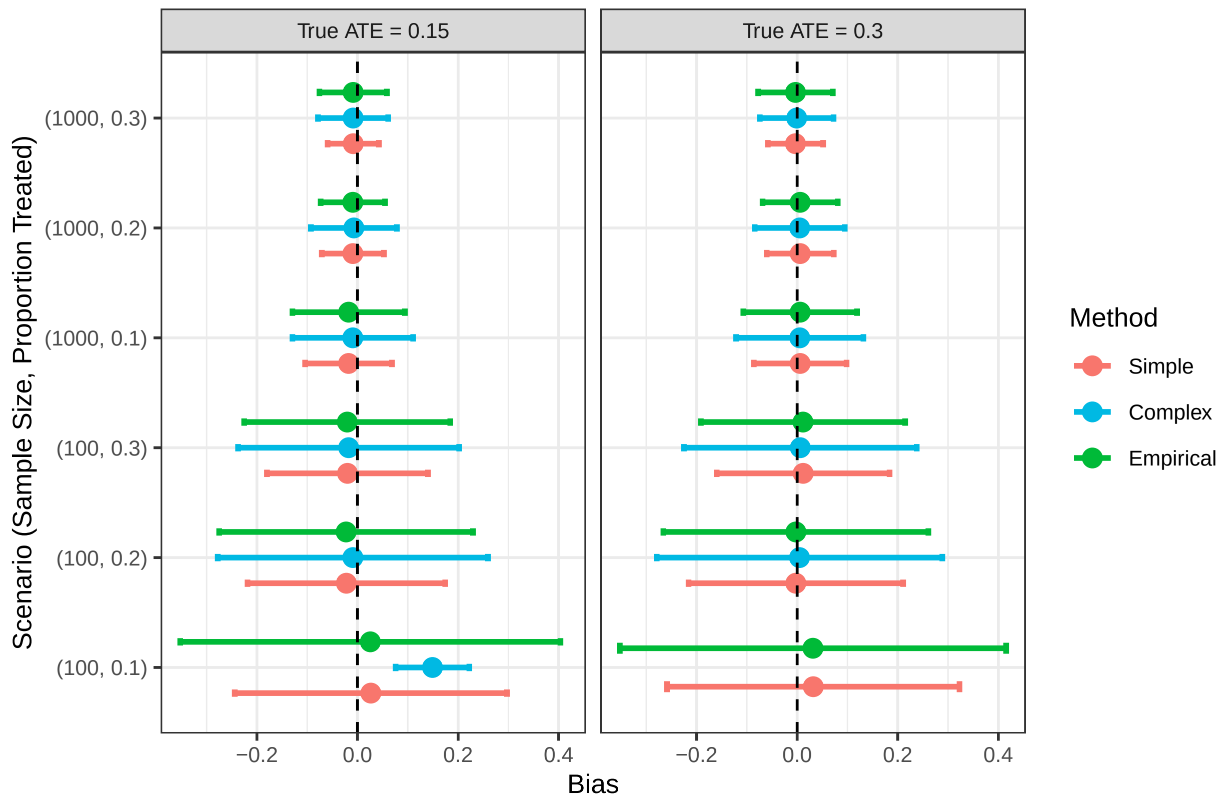
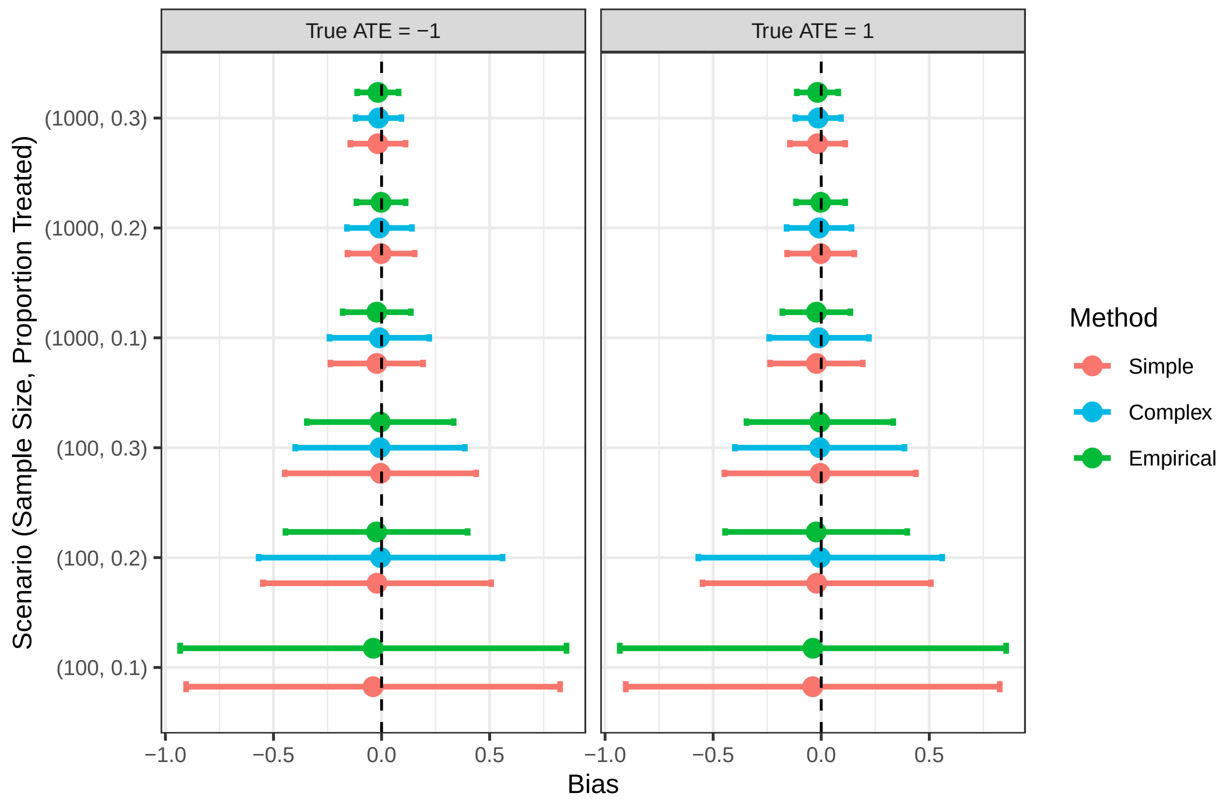


Figure 8: Continuous Bias Distributions



Additional analysis was performed on the bias demonstrated in each of the bootstrap methods and the empirical calculation. In Figures 6 and 7, the mean and standard deviation of the bias (the difference between the estimated and true average treatment effect) are shown for all three methods in the binary and continuous settings, respectively. The standard deviation of bias increases as the number of initial samples (n_{sample}) decreases. These results are summarized in detail in Tables 1 and 2.

The two tables below summarize the results of all the plots above (Figure 3 - Figure 8). These tables include the standard deviation and bias of the empirical samples, 100 sub-populations before performing the simple bootstrap (Figure 1).

Table 1: Binary outcome summary

Binary Outcome	Empirical		Simple			Complex		
Scenario	E_SE	E_Bias	S_SE	S_Bias	S_CR	C_SE	C_Bias	C_CR
Large Sample, ATE = 0.15, p = 0.1	**0.115**	**-0.029**	**0.054**	**-0.007**	**0.91**	**0.036**	**-0.008**	**1.00**
Large Sample, ATE = 0.15, p = 0.2	**0.071**	**0.004**	**0.043**	**-0.017**	**0.83**	**0.030**	**-0.015**	**0.99**
Large Sample, ATE = 0.15, p = 0.3	**0.052**	**-0.010**	**0.033**	**-0.014**	**0.88**	**0.028**	**-0.013**	**0.98**
Large Sample, ATE = 0.30, p = 0.1	**0.115**	**-0.029**	**0.061**	**-0.012**	**0.86**	**0.052**	**-0.005**	**0.99**
Large Sample, ATE = 0.30, p = 0.2	**0.071**	**0.004**	**0.047**	**-0.007**	**0.86**	**0.042**	**-0.010**	**0.95**
Large Sample, ATE = 0.30, p = 0.3	**0.052**	**-0.010**	**0.036**	**-0.009**	**0.89**	**0.032**	**-0.008**	**1.00**
Small Sample, ATE = 0.15, p = 0.1	**0.367**	**0.003**	**0.194**	**-0.034**	**0.80**	**NA**	**NA**	**NA**
Small Sample, ATE = 0.15, p = 0.2	**0.302**	**-0.038**	**0.140**	**0.008**	**0.86**	**0.109**	**-0.007**	**0.99**
Small Sample, ATE = 0.15, p = 0.3	**0.215**	**-0.010**	**0.115**	**-0.022**	**0.81**	**0.103**	**-0.013**	**0.98**
Small Sample, ATE = 0.30, p = 0.1	**0.367**	**0.003**	**0.202**	**-0.013**	**0.82**	**NA**	**NA**	**NA**
Small Sample, ATE = 0.30, p = 0.2	**0.302**	**-0.038**	**0.142**	**0.010**	**0.85**	**0.112**	**0.018**	**0.98**
Small Sample, ATE = 0.30, p = 0.3	**0.215**	**-0.010**	**0.113**	**-0.010**	**0.86**	**0.099**	**-0.008**	**0.99**

Table 2: Continuous outcome summary

Continuous Outcome	Empirical		Simple			Complex		
Scenario	E_SE	E_Bias	S_SE	S_Bias	S_CR	C_SE	C_Bias	C_CR
Large Sample, ATE = +1, p = 0.1	**0.115**	**-0.029**	**0.114**	**-0.028**	**0.94**	**0.054**	**-0.014**	**1.00**
Large Sample, ATE = +1, p = 0.2	**0.071**	**0.004**	**0.071**	**0.003**	**0.95**	**0.041**	**-0.010**	**1.00**
Large Sample, ATE = +1, p = 0.3	**0.052**	**-0.010**	**0.052**	**-0.010**	**0.97**	**0.034**	**-0.013**	**0.99**
Large Sample, ATE = -1, p = 0.1	**0.115**	**-0.029**	**0.114**	**-0.028**	**0.94**	**0.054**	**-0.014**	**1.00**
Large Sample, ATE = -1, p = 0.2	**0.071**	**0.004**	**0.071**	**0.003**	**0.95**	**0.041**	**-0.010**	**1.00**
Large Sample, ATE = -1, p = 0.3	**0.052**	**-0.010**	**0.052**	**-0.010**	**0.97**	**0.034**	**-0.013**	**0.99**
Small Sample, ATE = +1, p = 0.1	**0.367**	**0.003**	**0.367**	**0.002**	**0.94**	**NA**	**NA**	**NA**
Small Sample, ATE = +1, p = 0.2	**0.302**	**-0.038**	**0.302**	**-0.037**	**0.91**	**0.138**	**-0.030**	**1.00**
Small Sample, ATE = +1, p = 0.3	**0.215**	**-0.010**	**0.218**	**-0.006**	**0.97**	**0.115**	**-0.012**	**1.00**
Small Sample, ATE = -1, p = 0.1	**0.367**	**0.003**	**0.367**	**0.002**	**0.94**	**NA**	**NA**	**NA**
Small Sample, ATE = -1, p = 0.2	**0.302**	**-0.038**	**0.302**	**-0.037**	**0.91**	**0.138**	**-0.030**	**1.00**
Small Sample, ATE = -1, p = 0.3	**0.215**	**-0.010**	**0.218**	**-0.006**	**0.97**	**0.115**	**-0.012**	**1.00**

Discussion

In summary, the simple bootstrap method tended to have smaller standard error estimates compared to the complex bootstrap method overall and to underestimate the sampling variability of the treatment effect for binary outcomes. As we can see from our 95% confidence interval coverage rate plots, the simple bootstrap tended to have less conservative type I error rate than 0.05 whereas the complex bootstrap tended to have more conservative type I error rate than 0.05. As a result, using the complex bootstrap tended to decrease the statistical power. However, differences were less pronounced in settings with larger sample sizes and higher treatment prevalence. Based on the results of this study, it is recommended to use the simple bootstrap method for continuous data, as the complex bootstrap confidence intervals typically had statistically significantly high coverage rates. For binary data, it is recommended to use the complex bootstrap, particularly in settings with larger sample sizes. The simple bootstrap tended to produce statistically significantly low coverage rates for binary data. Notably, the complex bootstrap was not a reliable method in a setting with small sample size and low treatment prevalence (n = 100, 10% treatment).

Limitations

One drawback of this study was a lack of sample size plurality. It would be beneficial to understand the performance of these methods in settings with larger sample sizes (e.g., 5,000 or 10,000 subjects). This was unfortunately too computationally intensive for this study. Another limitation was the number of initial samples made. In the procedures above, a set of 100 initial samples was taken. Each sample resulted in one confidence interval, which was the primary method of assessing standard error estimates. With only 100 confidence intervals, the statistically insignificant range for coverage rate was larger than would be ideal. Running the same procedures with 1,000 initial samples would allow for a narrower range of acceptable coverage rates, yielding more power to detect differences between methods.

Future Work

Similar studies in the future could focus on improving some of the limitations mentioned above. In particular, working with a larger number of initial samples and larger sample sizes would give more insight into how well the simple and complex bootstrap methods estimate standard error. As detailed above, the data generation and treatment assignment models are relatively simple with standard normally distributed covariates. Covariate correlations were also held constant in each of the simulated scenarios. It is recommended that these methods be studied in settings with varying parametric or non-parametric data to further understand their performance.

Group Contributions

Waveley and Hun worked on binary code.

Tucker worked on continuous code.

Amy worked on figures and data gen.

Jimmy worked on the presentation and finer points on the report. Jimmy did math.

We all worked on the presentation and report.

We all attended meetings. Lots of meetings. Productive meetings!

References

1. Rosenbaum, P. R. und DB Rubin 1983."The Central Role of the Propensity Score in Observational Studies for Causal Effects". Biometrika, 70(1), 1-55
2. Ho, D., Imai, K., King, G., Stuart, E., & Whitworth, A. (2018). Package 'MatchIt'
3. Schafer, J. L., & Kang, J. (2008). Average causal effects from nonrandomized studies: a practical guide and simulated example. Psychological methods, 13(4), 279.
4. Austin, P. C., & Small, D. S. (2014). The use of bootstrapping when using propensity-score matching without replacement: a simulation study. Statistics in medicine, 33(24), 4306-4319.

Appendices

Appendix A: Seed Vector

```r
set.seed(20220217)

seed_vec <- runif(100000, min = 100, max = 99999999) %>% round(0) %>% unique()
```

Appendix B: Simple Bootstrap, Binary Outcome

```
simple_boot <- function(df, n = m_boot, size = df %>% pull(A) %>% sum(), seeds = seed_vec){
  boots <- list()
  for (i in 1:n) {
    set.seed(seeds[i])
    boots[[i]] <-
      df %>%
      filter(subclass %in% sample(levels(subclass),
                                  size,
                                  replace =  TRUE))
  }
  return(boots)
}
```

Appendix C: Complex Bootstrap, Binary Outcome

```r
generate_boots <- function(df, iter = m_boot, seeds = seed_vec){

  boot_samples <- list()
  matched_boot_df <- list()
  boot_results <- tibble()

  for (i in 1:iter) {
    set.seed(seeds[[i]])

    boot_samples[[i]] <- sample_n(df, nrow(df), replace = TRUE)

    matched <- matchit(A ~ L2 + L3, data = boot_samples[[i]],
                       distance = "glm", link = "logit",
                       method = "nearest", ratio = 1)

    matched_boot_df[[i]] <- match.data(matched, distance = "ps")
  }

  return(matched_boot_df)
}
```

Appendix D: Outcome Function, Binary Outcome

```r
outcome_model_list <- function(list) {
  tib_coef <- tibble()
  boots <- tibble(mean1 = NA,
                  mean0 = NA,
                  difference = NA)
  for (i in 1:length(list)) {
    mod <- glm(Y ~ A + ps,
               data = list[[i]],
               weights = weights,
               family = "binomial")

    sampl_all_treated <-
      list[[i]] %>%
      mutate(A = 1)

    sampl_all_untreated <-
      list[[i]] %>%
      mutate(A = 0)

    sampl_all_treated$pred.y <-
      predict(mod, sampl_all_treated, type = "response")

    sampl_all_untreated$pred.y <-
      predict(mod, sampl_all_untreated, type = "response")

    boots[i, "mean1"] <- mean(sampl_all_treated$pred.y)
    boots[i, "mean0"] <- mean(sampl_all_untreated$pred.y)
    boots[i, "difference"] <- boots[i, "mean1"] - boots[i, "mean0"]
  }
  return(boots)
}
```

Appendix E: Parameter Setting, Scenario 1

```
scenario_id <- 1

all_scenarios <- tibble(
  id = c(1:18),
  n_sample = c(rep(1000, 6), rep(10000, 6), rep(100, 6)),
  desired_prop = rep(c(0.1, 0.1, 0.2, 0.2, 0.3, 0.3),3),
  beta1 = rep(c(0.767,  1.386294), 9),
  beta0 = rep(c(0.422, -1.069315, 0.46, -1.138629, 0.499, -1.207944), 3)
)

desired_prop = all_scenarios %>% filter(id == scenario_id) %>% pull(desired_prop)
alpha1   = log(1.25)
alpha2   = log(1.75)
beta0    = all_scenarios %>% filter(id == scenario_id) %>% pull(beta0)
beta1    = all_scenarios %>% filter(id == scenario_id) %>% pull(beta1)
beta2    = log(1.75)
beta3    = log(1.25)
m_sample = 100
m_boot   = 500
n_sample = all_scenarios %>% filter(id == scenario_id) %>% pull(n_sample)
```