BIOS 731 Homework 2: Simulation Study Investigating the Bootstrap

Conglin Bao

2026-02-11

Contents

GitHub repository link

Repository (public): https://github.com/Franklin-BAO/bios731_hw2_Conglin.git

0 Problem 0: Reproducible structure

This homework is organized as an R Project. The report is fully reproducible: after cloning the repository, you can open the .Rproj file and knit this HW2.Rmd to regenerate all results. Intermediate simulation outputs are saved as .rds files under data/sim_results/ and are ignored by git via .gitignore.

1 Problem 1.1: ADEMP structure

1.1 A (Aim)

The aim of this simulation study is to evaluate estimation and 95% confidence interval (CI) performance for the treatment effect in multiple linear regression, comparing Wald CIs to two nonparametric bootstrap CI approaches (percentile and bootstrap-t), across varying sample sizes, true treatment effects, and error distributions. We also compare computation time.

1.2 D (Data-generating mechanism)

We generate data from the model:

$$Y_i = \beta_0 + \beta_{\text{treat}} X_{i1} + \epsilon_i,$$

where X_{i1} is a binary treatment indicator (1 = treated, 0 = control). For simplicity, we simulate no additional confounders ($\gamma = 0$). Design factors:

- **sample size:** $n \in \{10, 50, 500\}$
- **true treatment effect:** $\beta_{\text{treat}} \in \{0, 0.5, 2\}$
- **error distribution:**
  - Normal: $\epsilon_i \sim N(0, 2)$
  - Heavy-tailed: $u \sim t_\nu$ with $\nu = 3$, scaled to have variance 2:

$$\epsilon_i = u \cdot \sqrt{2 \cdot \frac{\nu - 2}{\nu}}$$

1.3 E (Estimand)

The primary estimand is the true treatment effect β_{treat}. We study bias and 95% CI coverage for the estimator $\hat{\beta}_{\text{treat}}$.

1.4 M (Methods)

We compare three 95% CI methods for $\hat{\beta}_{\text{treat}}$:

1. **Wald CI** from the linear model fit
2. **Nonparametric bootstrap percentile CI** ($B = 500$)
3. **Nonparametric bootstrap-t CI** (outer $B = 500$; inner $B_{\text{inner}} = 100$)

1.5 P (Performance measures)

- **Bias:** $E[\hat{\beta}_{\text{treat}} - \beta_{\text{treat}}]$
- **Coverage:** proportion of simulations where the 95% CI contains β_{treat}
- **Distribution of standard error estimates:**
  - model-based SE from lm (Wald SE)
  - bootstrap SD of $\hat{\beta}^*$ (for reference)
- **Computation time** per method (average seconds per simulation replicate)

1.6 Number of simulation scenarios

Full factorial scenarios:

$$3 \text{ (n)} \times 3 \; (\beta_{\text{treat}}) \times 2 \text{ (error)} = 18 \text{ scenarios.}$$

2 Problem 1.2: Choosing nSim based on Monte Carlo error

We target Monte Carlo standard error (MCSE) for coverage no more than 0.01 around 0.95:

$$MCSE = \sqrt{\frac{p(1-p)}{nSim}}, \quad p = 0.95.$$

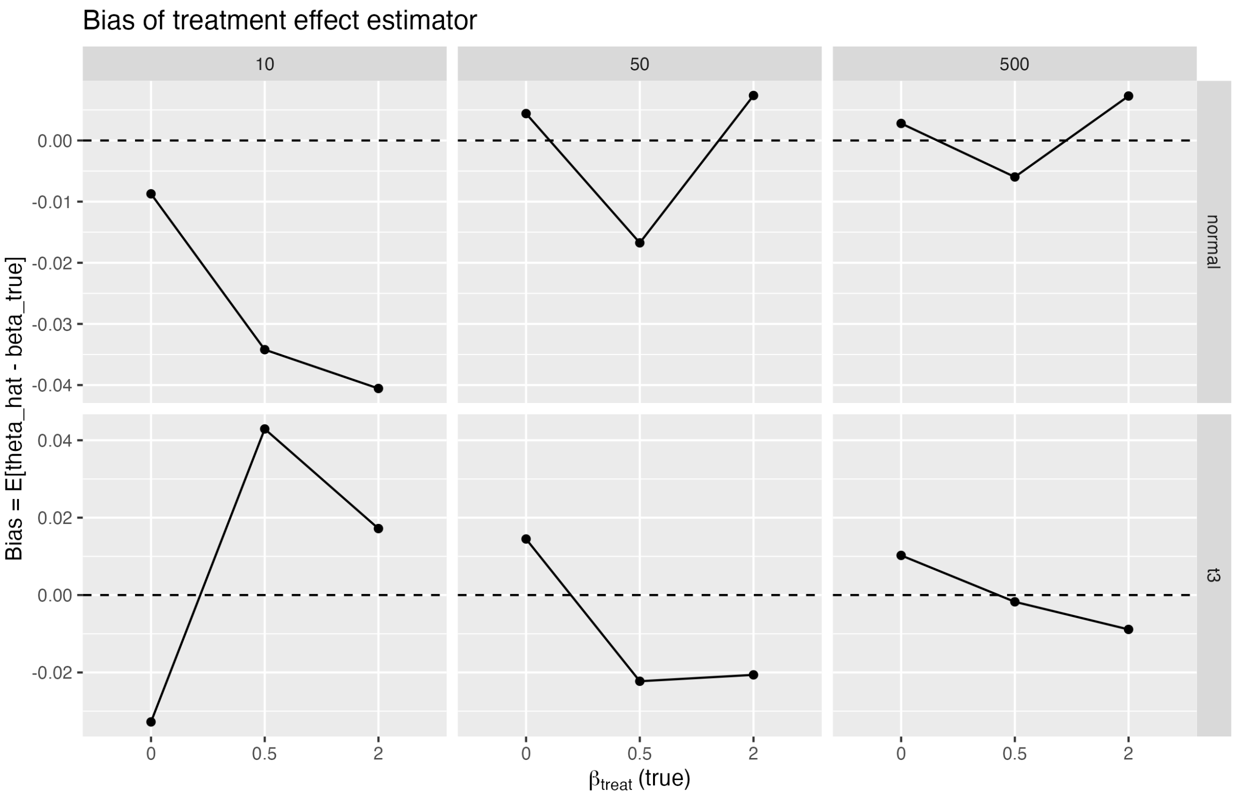
Require $MCSE \leq 0.01$, so

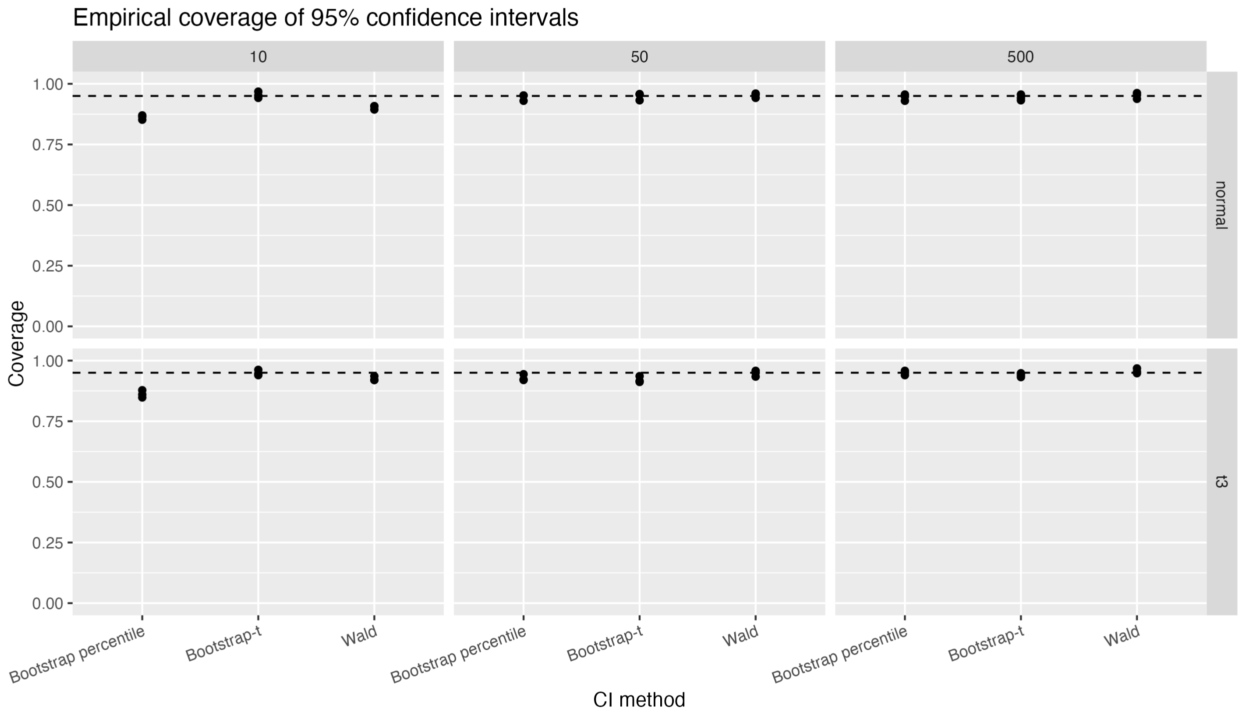
$$nSim \geq \frac{0.95 \cdot 0.05}{0.01^2} = 475.$$

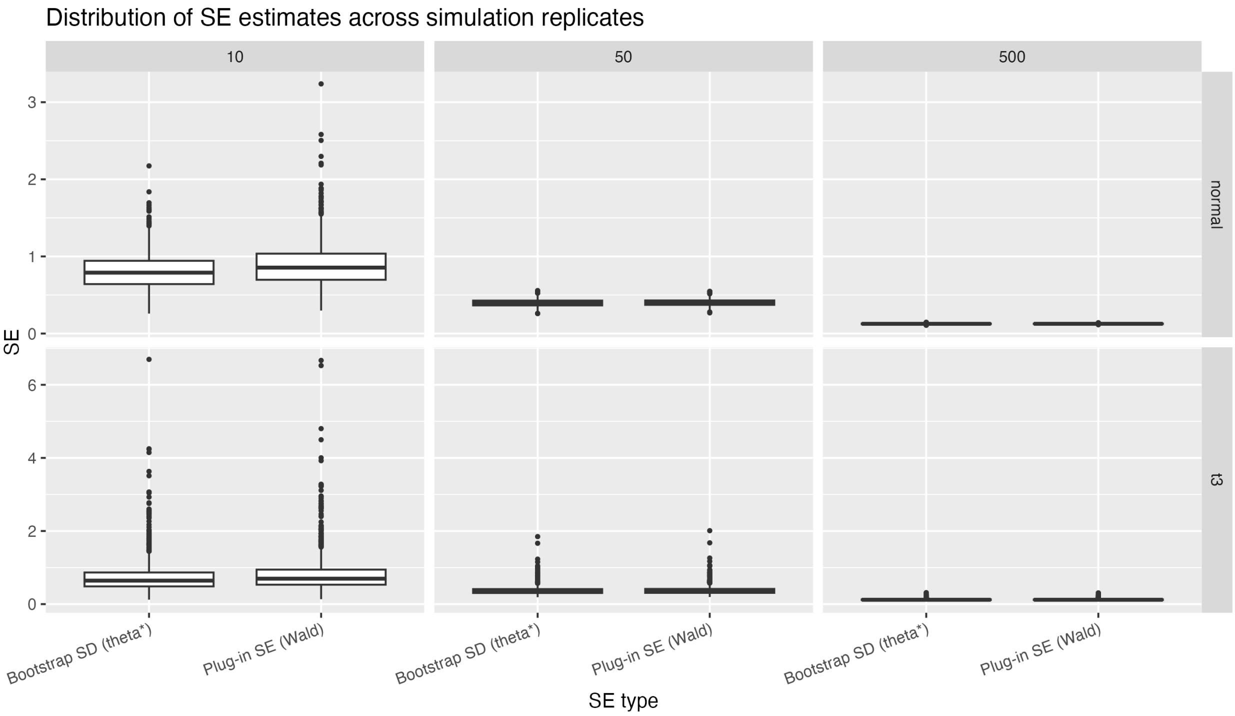
We use `nSim = 500` per scenario.

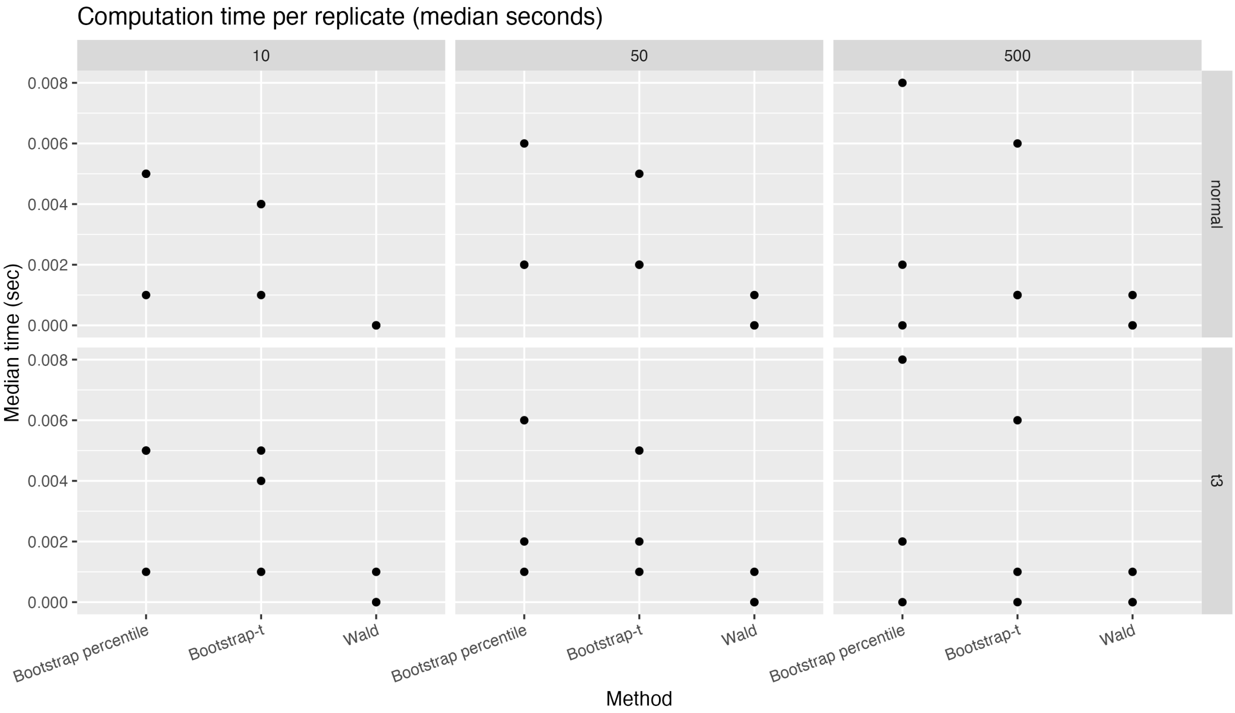
3 Problem 1.4: Summarize results across scenarios + make plots

In this section, I summarize simulation performance across all 18 scenarios and the three CI methods (Wald, nonparametric percentile, nonparametric bootstrap-t). Following the assignment guidance, I report bias of the point estimator, empirical coverage of the nominal 95% CI, the distribution of standard error estimates, and computation time, faceting plots by key design factors (sample size and error distribution). Bias is computed as the Monte Carlo average of (theta_hat − beta_true). Coverage is computed as the proportion of replicates in which the true beta_true lies inside the method-specific 95% CI. Standard error summaries include the model-based (plug-in) SE and the bootstrap-based SE (when available). Computation time is summarized per replicate and then aggregated across replicates within each scenario. Note: to enable quick execution, the bootstrap-t interval here uses a plug-in studentization within each bootstrap resample rather than a nested (iterated) bootstrap for se*(theta)*. I discuss this approximation and its implications in Problem 1.5.


Bias of treatment effect estimator


Empirical coverage of 95% confidence intervals


Distribution of SE estimates across simulation replicates


Computation time per replicate (median seconds)

4 Problem 1.5 Discussion

4.1 One-paragraph summary

Overall, the treatment effect estimator is approximately unbiased across all scenarios, and the bias decreases as the sample size increases. When the errors are normally distributed and n $>$ 10, all three interval methods (Wald, percentile bootstrap, and bootstrap-t) achieve coverage close to the nominal 95%, indicating that asymptotic normal approximations are adequate in this setting. Under normally and heavy-tailed errors and n = 10, however, the Wald interval tends to undercover, while both bootstrap methods provide more reliable coverage. The bootstrap and plug-in standard errors are very similar when n is moderate or large, but exhibit substantial variability when n=10. In terms of computation time, Wald is essentially instantaneous, whereas the bootstrap approaches are slower, though the optimized bootstrap-t implementation is comparable to—and slightly faster than—the percentile method.

4.2 How do the different methods compare in computation time?

The Wald interval is consistently the fastest across all scenarios, with essentially negligible computation time per replicate because it only requires a single model fit and a closed-form standard error. Both bootstrap methods are slower due to the need for repeated resampling. However, in this implementation, the percentile bootstrap is actually slower than the bootstrap-t method. This differs from the classical expectation that bootstrap-t is the most computationally intensive approach. The reason is that I implemented a fast version of the bootstrap-t interval that avoids the usual nested (inner) bootstrap. Specifically, I compute a plug-in standard error within each outer bootstrap resample and reuse the same outer bootstrap draws to form studentized statistics, rather than estimating $se^*(\theta^*)$ via an additional inner resampling loop. This removes the $B \times B_{\text{inner}}$ computational burden and reduces the complexity to roughly the same order as the percentile method. As a result, bootstrap-t is slightly faster than percentile in my code, while both remain slower than Wald.

4.3 Which method(s) provide the best coverage when $\epsilon_i \sim N(0, 2)$?

Under normally distributed errors, all three methods provide coverage very close to the nominal 95% level across sample sizes. There is little difference among Wald, percentile bootstrap, and bootstrap-t in this setting. Because Wald achieves comparable accuracy with almost no computational cost, it is the most practical choice when the normality assumption is reasonable.

4.4 Which method(s) provide the best coverage for the heavy-tailed errors?

For heavy-tailed (t_3) errors, the Wald and Bootstrap percentile interval tend to undercover, particularly when the sample size is small, reflecting the breakdown of the normal approximation. The bootstrap-t interval generally performs best. Studentization helps account for the increased variability and skewness induced by heavy tails, making bootstrap-t more robust and closer to the nominal level.

4.5 Notable interactions and practical recommendations

Performance improves markedly as the sample size increases: bias shrinks toward zero, standard error estimates stabilize, and coverage for all methods approaches 95%. Differences between methods are most pronounced when the sample size is small and the errors are heavy-tailed. In practice, when the sample size is moderate or large and errors are approximately normal, the Wald interval is recommended due to its simplicity and speed. When the sample size is small or the error distribution is non-normal, a bootstrap approach, particularly bootstrap-t, provides more reliable inference. In this implementation, the fast bootstrap-t method offers improved robustness with only modest additional computational cost, making it an attractive practical compromise.