BIOS 731 final analysis: A simulation study investigating the bootstrap

Xinyu Dong

2026-02-10

```r
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)

suppressPackageStartupMessages({
  library(here)
  library(dplyr)
  library(ggplot2)
  library(tidyr)
})
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```r
# Source the scripts used in this report
source(here("simulations", "simulation_code.R"))
source(here("simulations", "summarize_results.R"))
source(here("simulations", "make_plots.R"))
#source(here("source", "run_simulations.R"))
# the last one is way too slow...
```

Repository link

GitHub repository: https://github.com/xinyudong1129/bios731_hw2_Dong

Project organization

This project follows the listed folder structure:

- `analysis/` : final reproducible report
- `source/` : master scripts to execute the full simulation study
- `simulations/` : step-by-step scripts and reusable functions to complete the simulation and provide visualization and tabular results
- `data/` : saved intermediate results (scenario-wise `.rds` and combined `.rds`)
- `figures/` : visualization plots saved for this report

Problem 1.1: ADEMP Framework

Aim (A). The goal of this simulation study is to evaluate and compare the finite-sample performance of three methods for constructing 95% confidence intervals for the treatment effect $\beta_{\text{treatment}}$ in a linear regression model. In particular, we assess statistical accuracy (bias and coverage) and computational efficiency across a range of realistic data-generating scenarios.

Data-generating mechanism (D). Data are generated from the linear model

$$Y_i = \beta_0 + \beta_{\text{treatment}} X_{i1} + \varepsilon_i,$$

where $X_{i1} \in \{0, 1\}$ is a binary treatment indicator with $\Pr(X_{i1} = 1) = 0.5$. Errors ε_i follow either a normal distribution $\mathcal{N}(0, 2)$ or a heavy-tailed t_ν distribution with $\nu = 3$, scaled to have variance 2. The simulation varies the following factors in a full factorial design:

$$n \in \{10, 50, 500\}, \quad \beta_{\text{treatment}} \in \{0, 0.5, 2\}, \quad \varepsilon_i \sim \mathcal{N}(0, 2) \text{ or } t_3 \text{ (scaled)}.$$

Estimand (E). The primary estimand is the treatment effect $\beta_{\text{treatment}}$. Secondary quantities of interest include the standard error of $\hat{\beta}_{\text{treatment}}$ and the coverage probability of the associated 95% confidence intervals.

Methods (M). Three methods for constructing confidence intervals are compared:

1. Wald (model-based normal approximation) confidence intervals,
2. Nonparametric bootstrap percentile confidence intervals,
3. Nonparametric bootstrap-*t* confidence intervals (double bootstrap for standard errors).

All methods are implemented in the simulation code (see `simulations/simulation_code.R`) and orchestrated across scenarios in `source/run_simulation.R`.

Performance measures (P). Performance is summarized using:

- Bias of $\hat{\beta}_{\text{treatment}}$,
- Empirical coverage probability of 95% confidence intervals,
- Distribution of estimated standard errors,
- Computation time for each method.

Number of scenarios. The full factorial design includes $3 \times 3 \times 2 = 18$ unique simulation scenarios.

Problem 1.2: Choice of n_{sim}

Problem 1.2: Choice of n_{sim}

To estimate a nominal coverage probability of 0.95 with Monte Carlo error no larger than 0.01, we use the standard approximation

$$\text{MCSE}(\hat{p}) \approx \sqrt{\frac{p(1-p)}{n_{\text{sim}}}},$$

where $p \approx 0.95$. Solving

$$\sqrt{\frac{0.95 \times 0.05}{n_{\text{sim}}}} \leq 0.01$$

yields

$$n_{\text{sim}} \geq \frac{0.95 \times 0.05}{(0.01)^2} \approx 475.$$

Accordingly, we set $n_{\text{sim}} = 500$ Monte Carlo replications for each simulation scenario throughout the study.

Problem 1.3: Bootstrap-t Implementation

Problem 1.3: Bootstrap-t Implementation

For the bootstrap-t method, we approximate the distribution of the pivotal quantity

$$t^* = \frac{\hat{\theta}^* - \hat{\theta}}{\widehat{\text{se}}^*},$$

where $\hat{\theta}$ denotes the original estimate of $\beta_{\text{treatment}}$, $\hat{\theta}^*$ is the estimate from an outer bootstrap sample, and $\widehat{\text{se}}^*$ is the corresponding standard error estimated via an inner bootstrap procedure. Specifically, for each outer bootstrap replicate $b = 1, \dots, B$, we:

1. Resample the data with replacement to obtain a bootstrap sample and compute $\hat{\theta}^*_b$.
2. Perform an inner bootstrap with B_{inner} resamples to estimate $\widehat{\text{se}}^*_b$.
3. Compute $t^*_b = (\hat{\theta}^*_b - \hat{\theta})/\widehat{\text{se}}^*_b$.

Let $q^*_{\alpha/2}$ and $q^*_{1-\alpha/2}$ denote the empirical quantiles of $\{t^*_b\}_{b=1}^B$. The bootstrap-t confidence interval is then constructed as

$$\left[\hat{\theta} - q^*_{1-\alpha/2}\,\widehat{\text{se}},\ \hat{\theta} - q^*_{\alpha/2}\,\widehat{\text{se}}\right],$$

where $\widehat{\text{se}}$ is the model-based standard error from the original sample. We use $B = 500$ outer bootstrap replicates and $B_{\text{inner}} = 100$ inner bootstrap replicates.

However, given limited computational power of my laptop,we set $B = 50$ outer bootstrap replicates and $B_{inner} = 100$ inner bootstrap replicates.

Problem 1.4: Results Summary

All numerical results and figures in this section are generated by the scripts `source/summarize_results.R` and `source/make_plots.R` using the simulation output saved to the `data/` directory.

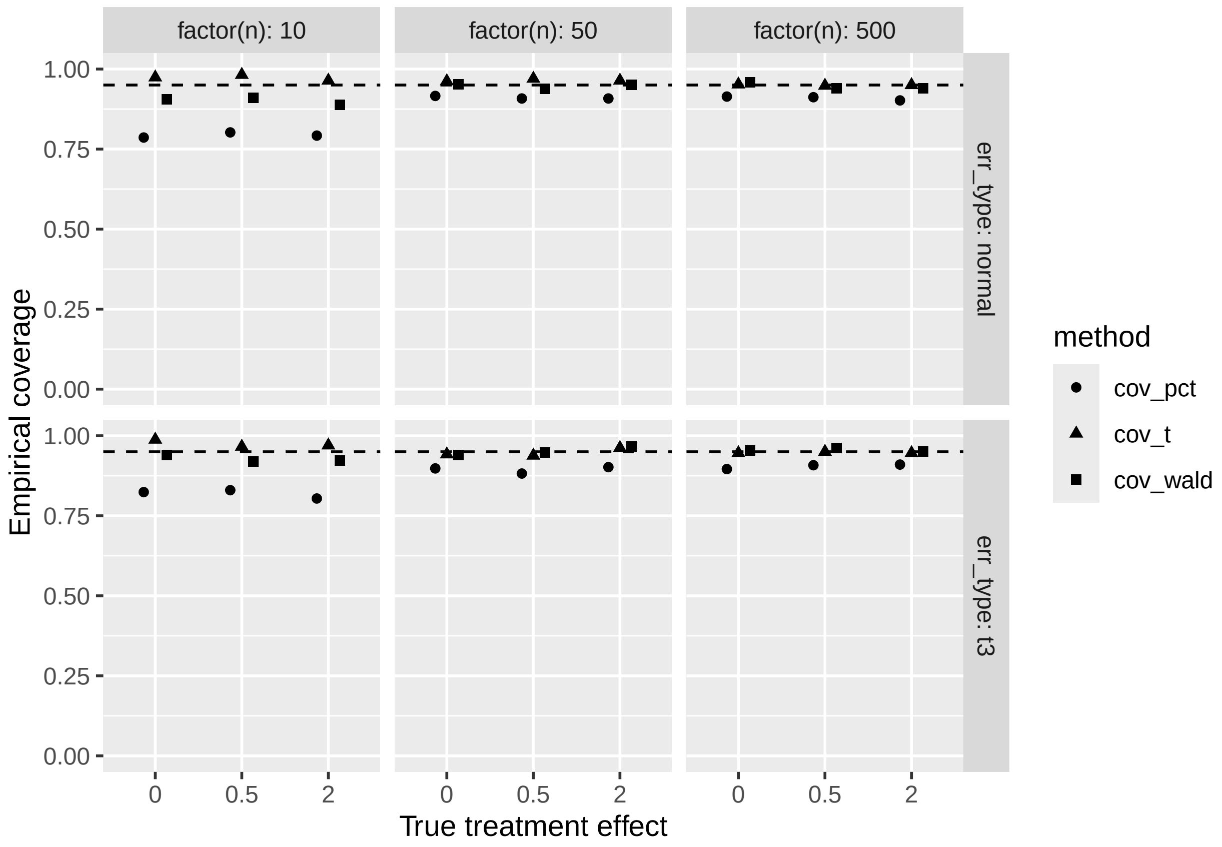
```
res <- readRDS(here("data", "combined_results.rds"))
head(res)
```

```
## # A tibble: 6 x 15
##   beta_hat se_hat ci_wald_l ci_wald_u ci_pct_l ci_pct_u ci_t_l ci_t_u time_wald
##      <dbl>  <dbl>     <dbl>     <dbl>    <dbl>    <dbl>  <dbl>  <dbl>     <dbl>
## 1   1.23    1.11     -0.946    3.41      0.467    1.93   -2.07   3.34      0
## 2   0.646   0.808    -0.938    2.23     -0.627    2.31   -1.10   3.23      0.0200
## 3  -1.65    0.856    -3.33     0.0240   -3.07    -0.406  -6.45   0.648     0
## 4   0.557   0.831    -1.07     2.19     -0.277    1.74   -1.58   3.65      0
## 5   0.995   0.846    -0.664    2.65     -0.656    2.46   -2.92   6.63      0
## 6   0.0624  0.971    -1.84     1.97     -1.35     1.30   -3.84   3.34      0
## # i 6 more variables: time_pct <dbl>, time_t <dbl>, n <dbl>, beta_trt <dbl>,
## #   err_type <chr>, scenario_id <chr>
```

include our plots:

1. coverage probability

```
p_cov <- plot_coverage(res)
ggsave(
  filename = here("figures", "coverage_by_method.png"),
  plot = p_cov,
  width = 8, height = 5, dpi = 300
)
p_cov
```

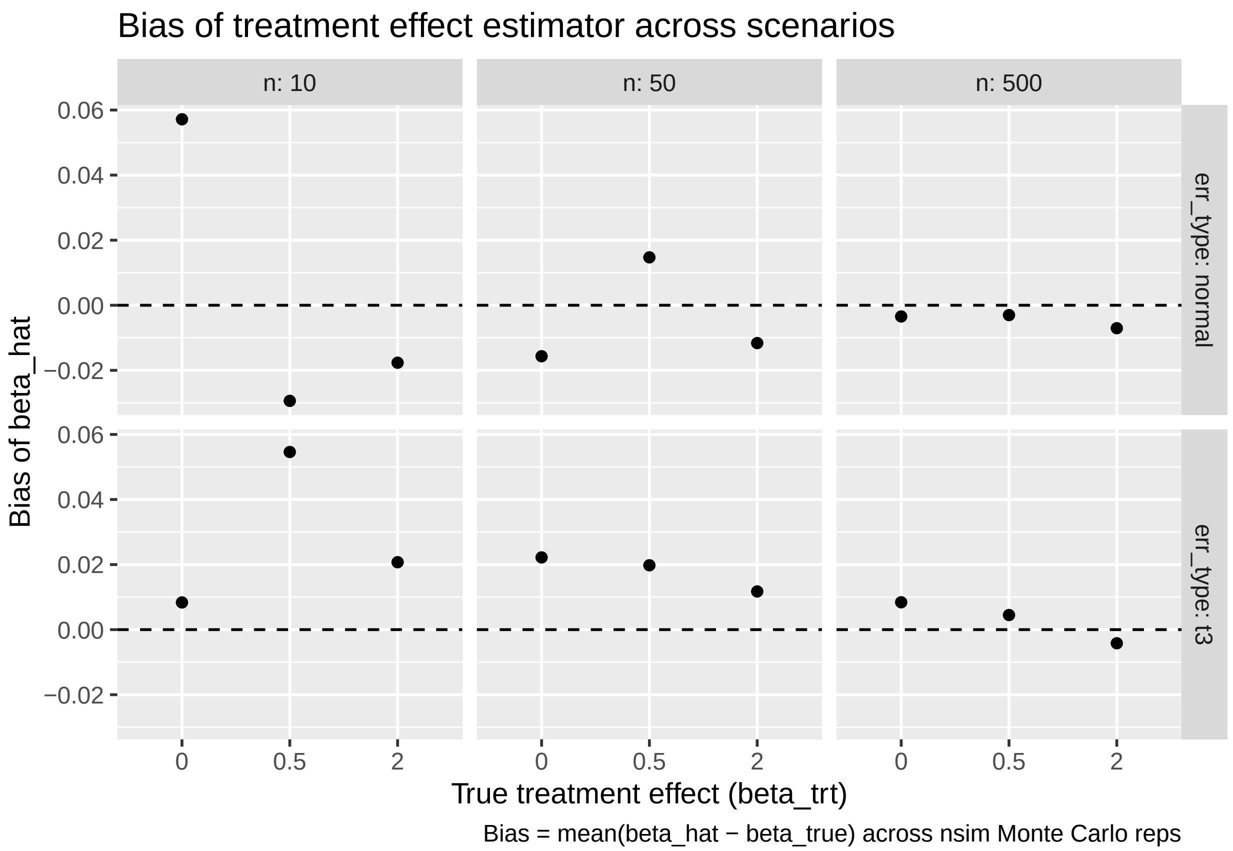


Empirical Coverage of 95% Confidence Intervals

This figure shows the empirical coverage probabilities of nominal 95% confidence intervals for the treatment effect across sample sizes $n \in \{10, 50, 500\}$, true effects $\beta_{\text{trt}} \in \{0, 0.5, 2\}$, and error distributions (normal vs. t_3). The dashed horizontal line indicates the nominal level 0.95. For normal errors, all methods achieve near-nominal coverage when $n \geq 50$. At $n = 10$, the Wald and percentile bootstrap intervals exhibit noticeable undercoverage, particularly for nonzero treatment effects. Under heavy-tailed errors, undercoverage is more pronounced for Wald and percentile methods, whereas the bootstrap-t interval remains closest to the nominal level. As n increases, the coverage of all methods converges toward 0.95, indicating asymptotic validity.

2. bias

```r
p_bias <- plot_bias(res)
ggsave(
  filename = here("figures", "bias_by_scenario.png"),
  plot = p_bias,
  width = 8, height = 5, dpi = 300
)
p_bias
```



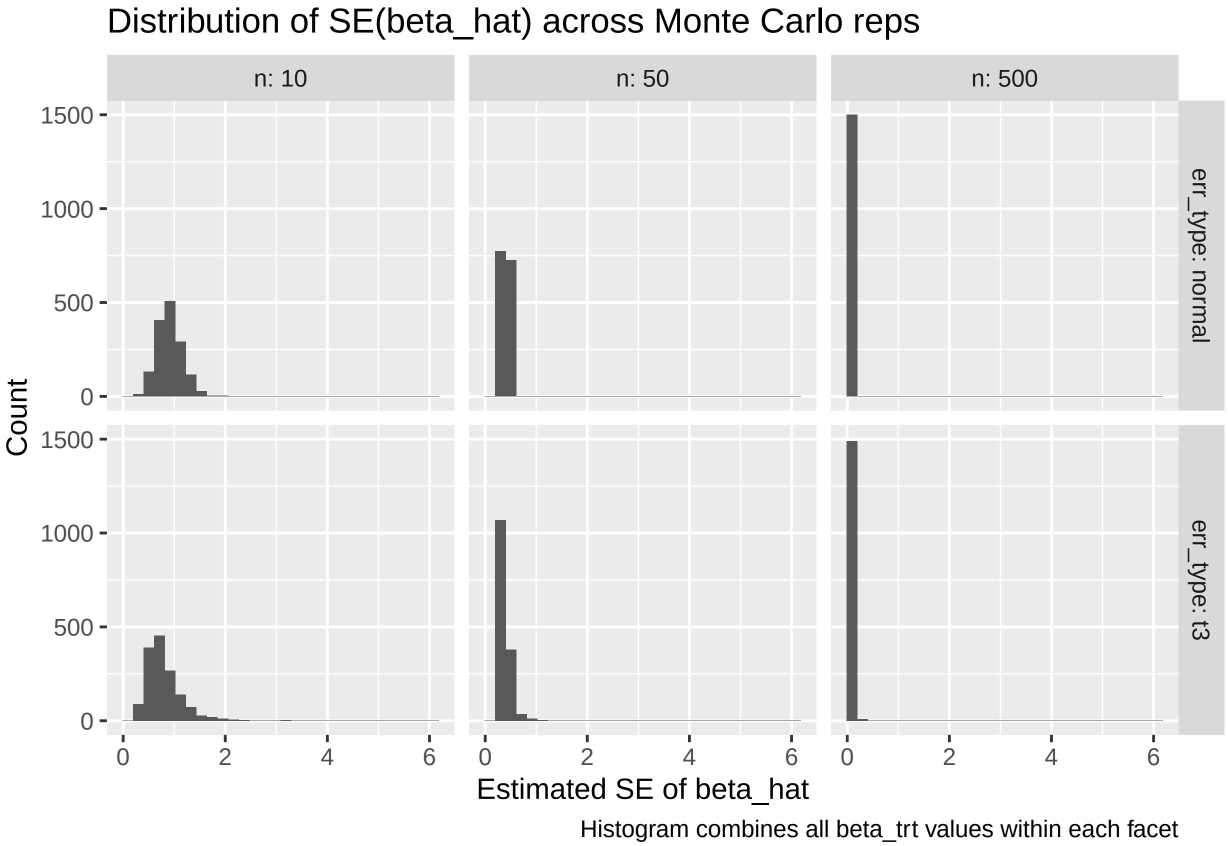
Bias of the Treatment Effect Estimator

This figure reports the Monte Carlo bias of the estimator, defined as $\mathbb{E}(\hat{\beta} - \beta_{\text{trt}})$. Across all scenarios, the estimator is approximately unbiased, with bias values close to zero. Small-sample bias is more visible at

$n = 10$, particularly under heavy-tailed errors, but the magnitude is modest. As the sample size increases to $n = 50$ and $n = 500$, the bias shrinks toward zero under both error distributions. No systematic trend in bias is observed across values of β_{trt}, indicating that the linear regression estimator is centered around the true parameter.

3. estimated se of beta_hat

```
p_se <- plot_se_dist(res)
ggsave(
  filename = here("figures", "se_distribution.png"),
  plot = p_se,
  width = 8, height = 5, dpi = 300
)
p_se
```

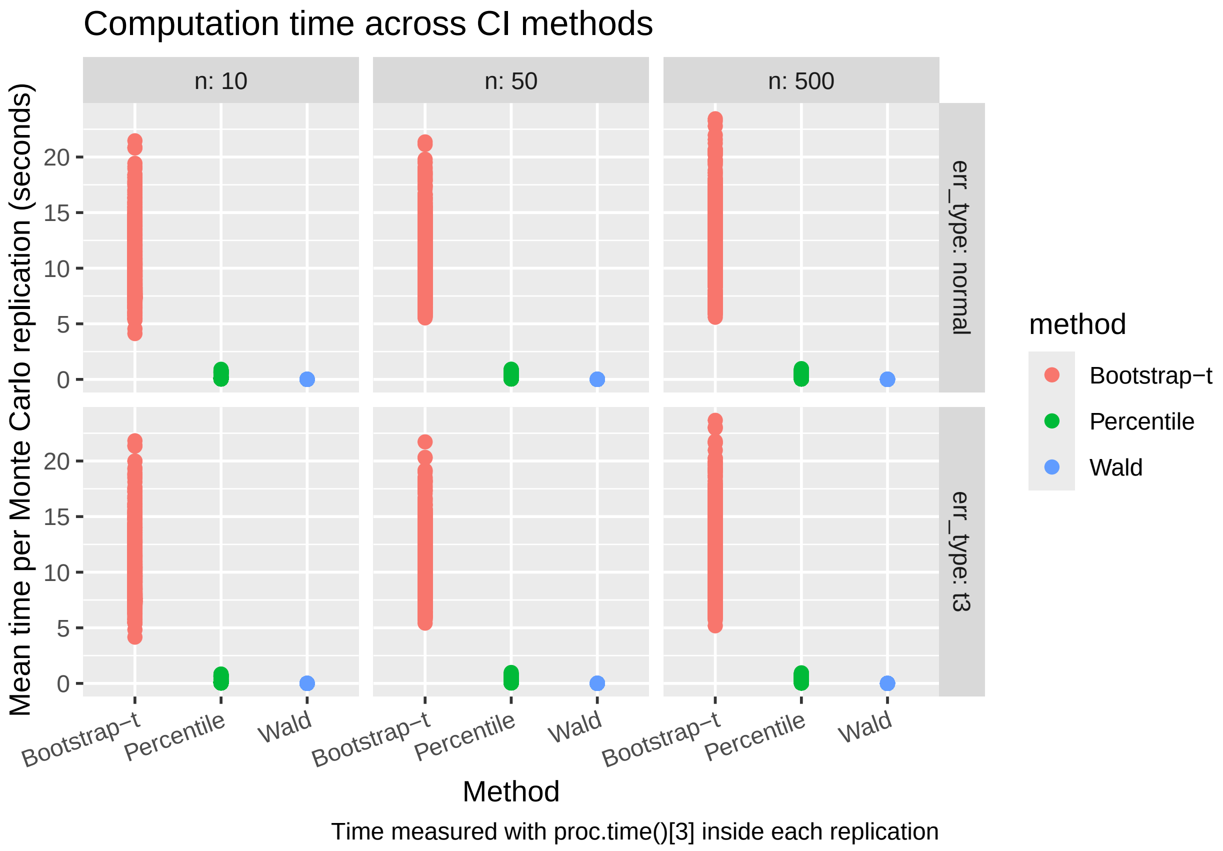


Distribution of Estimated Standard Errors

This figure displays the distribution of estimated standard errors of $\hat{\beta}$ across Monte Carlo replications. For $n = 10$, the distribution is wide and right-skewed, reflecting substantial variability in uncertainty estimates, particularly under heavy-tailed errors. As the sample size increases to $n = 50$ and $n = 500$, the distributions become increasingly concentrated, indicating stabilization of the variance estimates. The heavier dispersion under t_3 errors highlights the impact of heavy tails on estimator variability in small samples. These patterns are consistent with the expected $n^{-1/2}$ convergence of standard errors.

4. Computation Time

```
p_time <- plot_time(res)
ggsave(
  filename = here("figures", "time_by_method.png"),
  plot = p_time,
  width = 8, height = 5, dpi = 300
)
p_time
```



Computation Time Across Methods

This figure compares the mean computation time per Monte Carlo replication for the Wald, bootstrap percentile, and bootstrap-t confidence intervals. The Wald interval is computationally negligible across all scenarios. The percentile bootstrap incurs moderate computational cost due to resampling, while the bootstrap-t method is substantially more expensive because of the nested (double) bootstrap procedure. Computation time increases mildly with sample size, but the dominant factor is the resampling structure rather than n. This highlights a clear trade-off between finite-sample robustness and computational efficiency.

Summary

Overall, the bootstrap-t method provides the most reliable finite-sample coverage, particularly under heavy-tailed errors, at the cost of substantially increased computation time. The Wald and percentile bootstrap intervals perform adequately in moderate to large samples but may undercover in small samples, especially when the error distribution is heavy-tailed.

Problem 1.5: Discussion

Overall, the simulation study demonstrates clear trade-offs between statistical accuracy and computational cost among the three confidence interval methods. Wald intervals are computationally efficient but can exhibit under-coverage in small samples and under heavy-tailed error distributions. The bootstrap percentile method improves robustness in moderate sample sizes but still shows sensitivity to heavy tails. The bootstrap-t method provides the most reliable coverage, particularly under heavy-tailed errors, at the expense of substantially higher computational cost due to the nested bootstrap procedure.

In terms of computation time, Wald intervals are orders of magnitude faster than either bootstrap method, while bootstrap-t intervals are the most computationally intensive. Under normally distributed errors, all methods perform reasonably well for moderate to large n, though bootstrap-t tends to achieve coverage closest to the nominal level. Under heavy-tailed errors, bootstrap-t provides the most accurate coverage, especially for small to moderate sample sizes. Notable interactions include improved performance of all methods as n increases and increased relative advantage of bootstrap-based methods under heavy-tailed error distributions. Practically, these results suggest that while Wald intervals may suffice in large-sample, light-tailed settings, bootstrap-t intervals are preferable in small-sample or heavy-tailed contexts when computational resources permit.