Homework 1

Github link: https://github.com/xxou617/bios731_hw1_ou

$$Y_i = \beta_0 + \beta_{treatment}X_{i1} + \mathbf{Z_i}^T\boldsymbol{\gamma} + \epsilon_i$$

Notation is defined below:

- Y_i: continuous outcome
- X_{i1}: treatment group indicator; $X_{i1} = 1$ for treated
- $\mathbf{Z_i}$: vector of potential confounders
- $\beta_{treatment}$: average treatment effect, adjusting for $\mathbf{Z_i}$
- $\boldsymbol{\gamma}$: vector of regression coefficient values for confounders
- ϵ_i: errors, we will vary how these are defined

In our simulation, we want to

- Estimate $\beta_{treatment}$ and $se(\hat{\beta}_{treatment})$
    - Evaluate $\beta_{treatment}$ through bias and coverage
    - We will use 3 methods to compute $se(\hat{\beta}_{treatment})$ and coverage:
        1. Wald confidence intervals (the standard approach)
        2. Nonparametric bootstrap percentile intervals
        3. Nonparametric bootstrap t intervals
    - Evaluate computation times for each method to compute a confidence interval
- Evaluate these properties at:
    - Sample size $n \in \{10, 50, 500\}$
    - True values $\beta_{treatment} \in \{0, 0.5, 2\}$
    - True ϵ_i normally distributed with $\epsilon_i \sim N(0, 2)$
    - True ϵ_i coming from a right skewed distribution
        * **Hint**: try $\epsilon_i \sim logNormal(0, \log(2))$
- Assume that there are no confounders ($\boldsymbol{\gamma} = 0$)
- Use a full factorial design

Problem 1.1 ADEMP Structure

Answer the following questions:

(1) How many simulation scenarios will you be running?

- 3 $\beta_{treatment}$
- 3 sample sizes
- 3 methods

- 2 error term distribution

$$3 * 3 * 2 * 3 = 54$$

Thus, we have 54 scenarios in total.

(2) What are the estimand(s)

- $\beta_{treatment}$

(3) What method(s) are being evaluated/compared?

- Wald confidence intervals (the standard approach), nonparametric bootstrap percentile intervals and nonparametric bootstrap t intervals for $se(\hat{\beta}_{treatment})$ in different scenario will be evaluated.

(4) What are the performance measure(s)?

- bias
- $se(\hat{\beta})$
- coverage
- computation time

Problem 1.2 nSim

Based on desired coverage of 95% with Monte Carlo error of no more than 1%, how many simulations (n_{sim}) should we perform for each simulation scenario? Implement this number of simulations throughout your simulation study.

$$n_{sim} = \frac{0.95(1 - 0.95)}{(0.01)^2} = 475$$

Problem 1.3 Implementation

We will execute this full simulation study. For full credit, make sure to implement the following:

- Well structured scripts and subfolders following guidance from `project_organization` lecture
- Use relative file paths to access intermediate scripts and data objects
- Use readable code practices
- Parallelize your simulation scenarios
- Save results from each simulation scenario in an intermediate `.Rda` or `.rds` dataset in a `data` subfolder

  - Ignore these data files in your `.gitignore` file so when pushing and committing to GitHub they don't get pushed to remote

- Make sure your folder contains a Readme explaining the workflow of your simulation study

  - should include how files are executed and in what order

- Ensure reproducibility! I should be able to clone your GitHub repo, open your `.Rproj` file, and run your simulation study

Problem 1.4 Results summary

Create a plot or table to summarize simulation results across scenarios and methods for each of the following.

- Bias of $\hat{\beta}$
- Coverage of $\hat{\beta}$
- Distribution of $se(\hat{\beta})$
- Computation time across methods

```r
library(tidyverse)
library(gt)
```

```r
load(here::here("data", "all_scenarios.Rdata"))
```

```r
# summarise results grouped by the scenarios

final_results <- final_results |>
  mutate(
    beta_true = as.factor(beta_true),
    epsilon_distr = factor(epsilon_distr, levels = c("Normal", "logNormal")),
    CI_method = factor(CI_method, levels = c("Wald", "boot quantile", "boot t"))
  )

simu_evaluate <- final_results |>
  group_by(n, beta_true, epsilon_distr, CI_method) |>
  summarise(
    bias = mean(error),
    avg_se = mean(est_se),
    avg_time = mean(cal_CI_time),
    coverage_rate = mean(coverage),
    .groups = "drop"
  )

head(simu_evaluate)
```

```
## # A tibble: 6 x 8
##        n beta_true epsilon_distr CI_method      bias avg_se avg_time coverage_rate
##    <dbl> <fct>     <fct>         <fct>         <dbl>  <dbl>    <dbl>         <dbl>
## 1     10 0         Normal        Wald       -0.0512   1.29   0.0190         0.909
## 2     10 0         Normal        boot qua~  -0.0512   1.15   0.351          0.857
## 3     10 0         Normal        boot t     -0.0512   1.29  61.8            0.981
## 4     10 0         logNormal     Wald       -0.00809  2.52   0.0194         0.987
## 5     10 0         logNormal     boot qua~  -0.00809  0.758  0.676          0.811
## 6     10 0         logNormal     boot t     -0.00809  2.52 118.             1
```
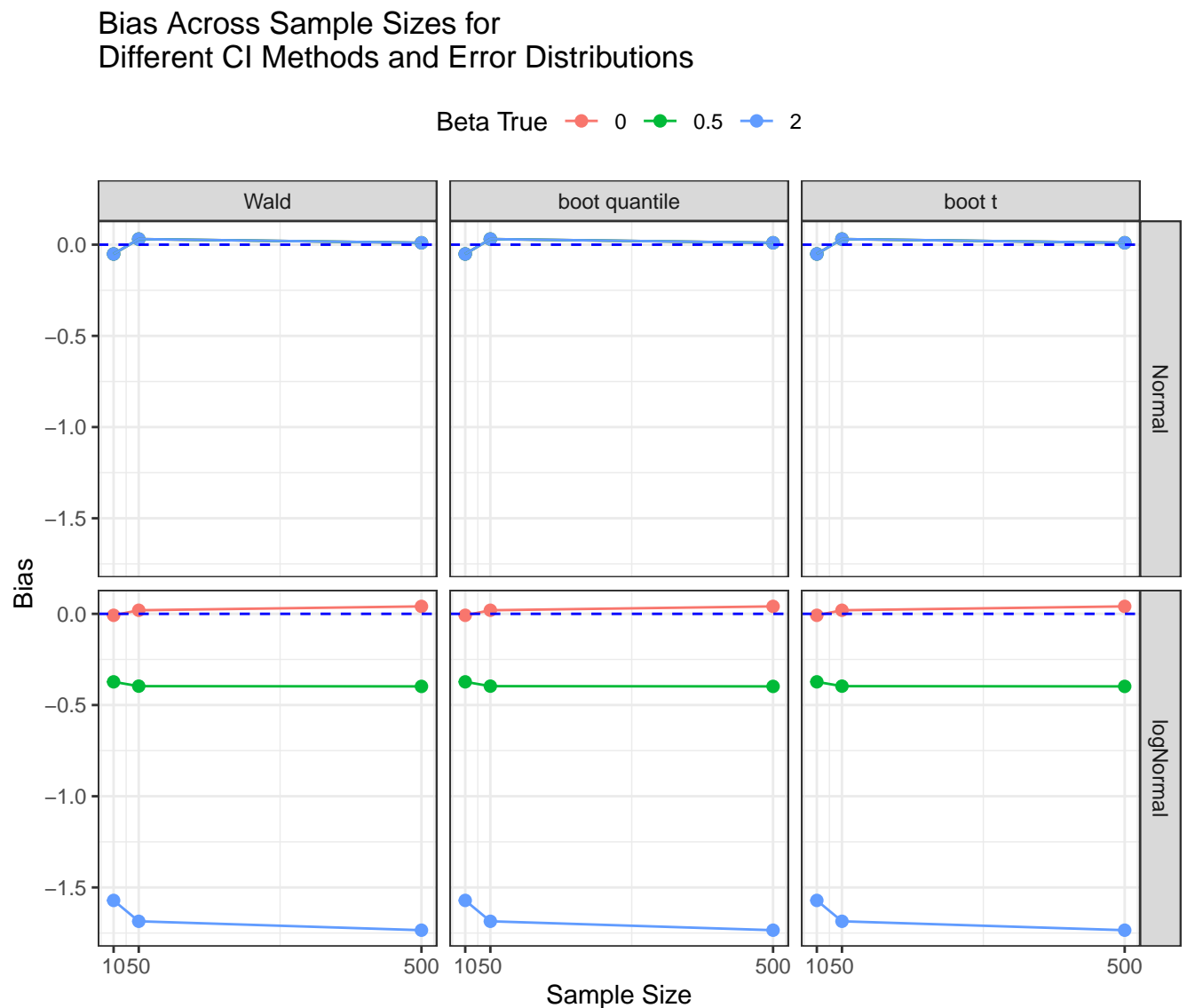
```r
sample_size = unique(simu_evaluate$n)
```

```r
# Plot for bias
ggplot(simu_evaluate,
       aes(x = n, y = bias, color = beta_true, group = beta_true)) +
  geom_point(size = 2) +
```

```
  geom_line() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "blue") +
  labs(x = "Sample Size", y = "Bias", color = "Beta True",
       title = "Bias Across Sample Sizes for \nDifferent CI Methods and Error Distributions") +
  theme_bw() +
  theme(legend.position = "top") +
  facet_grid( cols = vars(CI_method), rows = vars(epsilon_distr)) +
  scale_x_continuous(breaks = sample_size)  -> plot_bias

plot_bias
```


Bias Across Sample Sizes for
Different CI Methods and Error Distributions

```
# plot for coverage

ggplot(simu_evaluate,
       aes(x = n, y = coverage_rate, color = beta_true, group = beta_true)) +
  geom_point(size = 2) +
```

```
geom_line() +
geom_hline(yintercept = 0.95, linetype = "dashed", color = "blue") +
labs(x = "Sample Size", y = "Coverage", color = "Beta True",
     title = "Coverage rate Across Sample Sizes for \nDifferent CI Methods and Error Distributions")
theme_bw() +
theme(legend.position = "top") +
facet_grid( cols = vars(CI_method), rows = vars(epsilon_distr)) +
scale_x_continuous(breaks = sample_size)  -> plot_coverage

plot_coverage
```


Coverage rate Across Sample Sizes for
Different CI Methods and Error Distributions
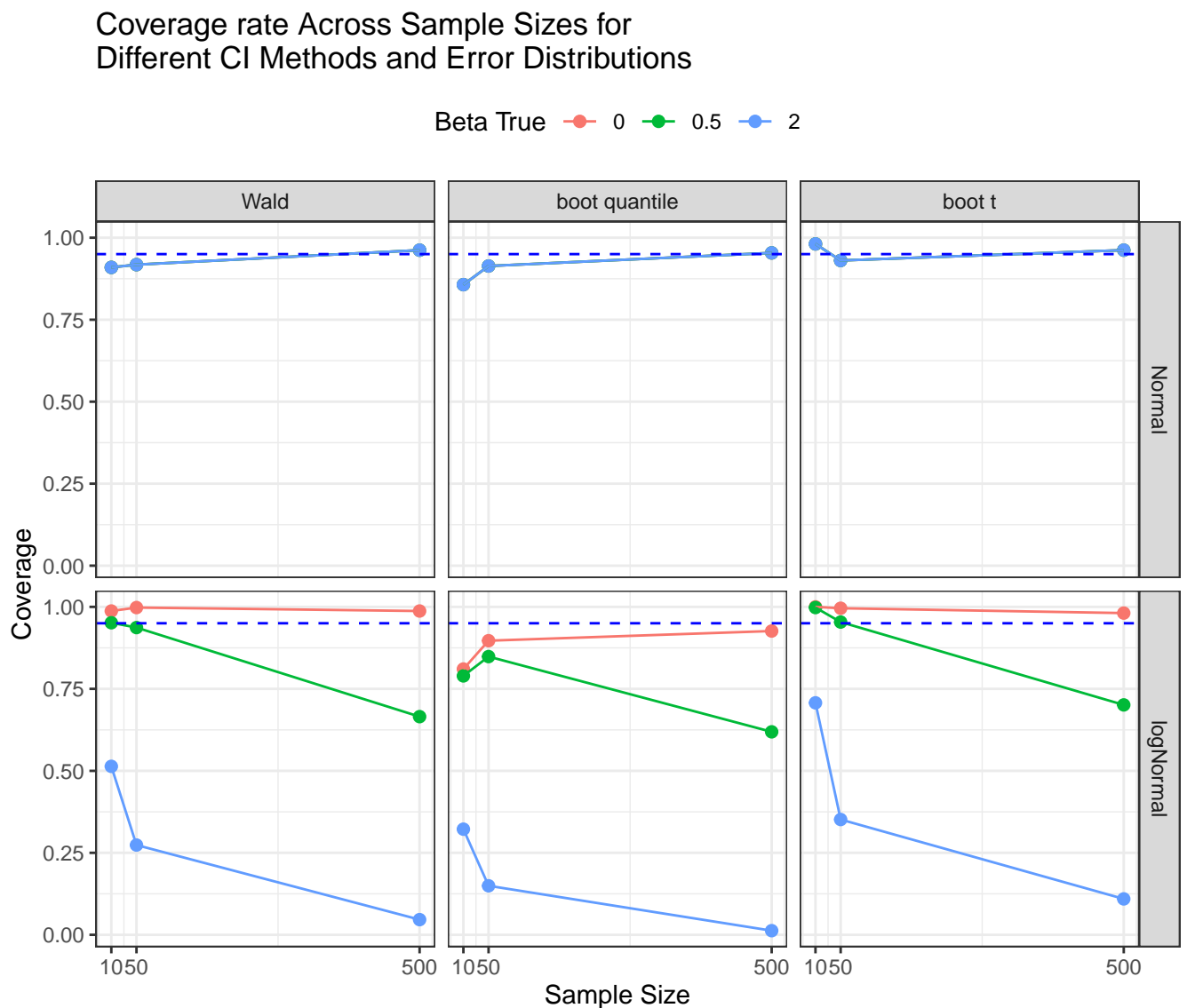
```
# distribution of se(beta_hat)
ggplot(final_results,
       aes(x = as.factor(n), y = est_se, color = beta_true)) +
  geom_boxplot() +
  labs(x = "Sample Size",
```

```
        y = "Estimated Standard Error",
        color = "Beta True",
        title = "Distribution of Estimated Standard Errors Across \nSample Sizes, CI Methods, and Error
    theme_bw() +
    theme(legend.position = "top") +
    facet_grid(cols = vars(CI_method), rows = vars(epsilon_distr), scales = "free_y") +
    scale_x_discrete(labels = sample_size) -> plot_se

plot_se
```

Distribution of Estimated Standard Errors Across Sample Sizes, CI Methods, and Error Distributions
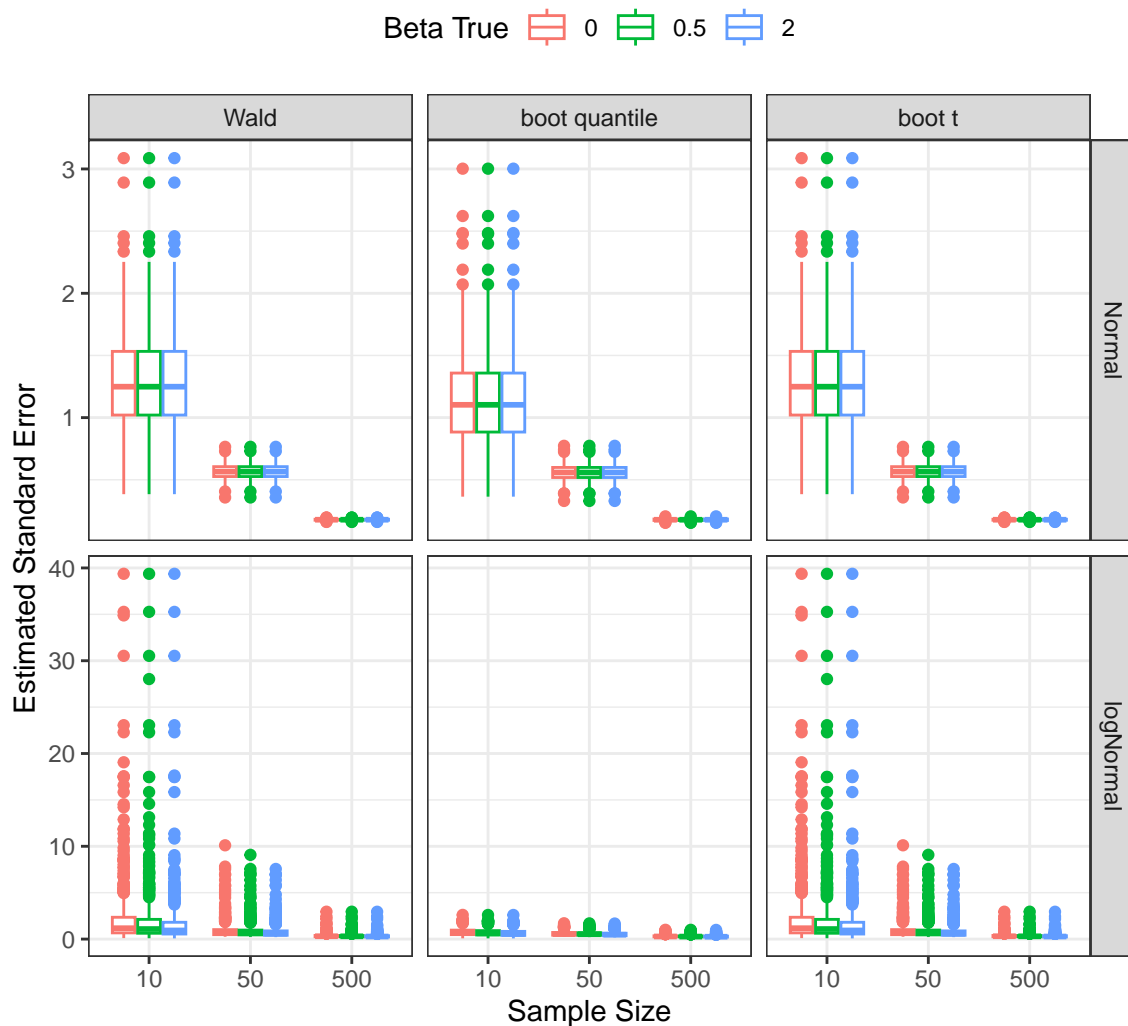


```
# plot for calculation time
ggplot(final_results, aes(x = as.factor(n), y = cal_CI_time, color = beta_true)) +
  geom_boxplot() +
  labs(x = "Sample Size",
       y = "Computation Time",
       color = "Beta True",
       title = "Distribution of Time for CI Calculation Across \nSample Sizes, CI Methods, and Error Dis
```

```
  theme_bw() +
  theme(legend.position = "top") +
  facet_wrap(~epsilon_distr+CI_method, scales = "free_y") +
  scale_x_discrete(labels = sample_size) -> plot_time

plot_time
```

Distribution of Time for CI Calculation Across Sample Sizes, CI Methods, and Error Distributions
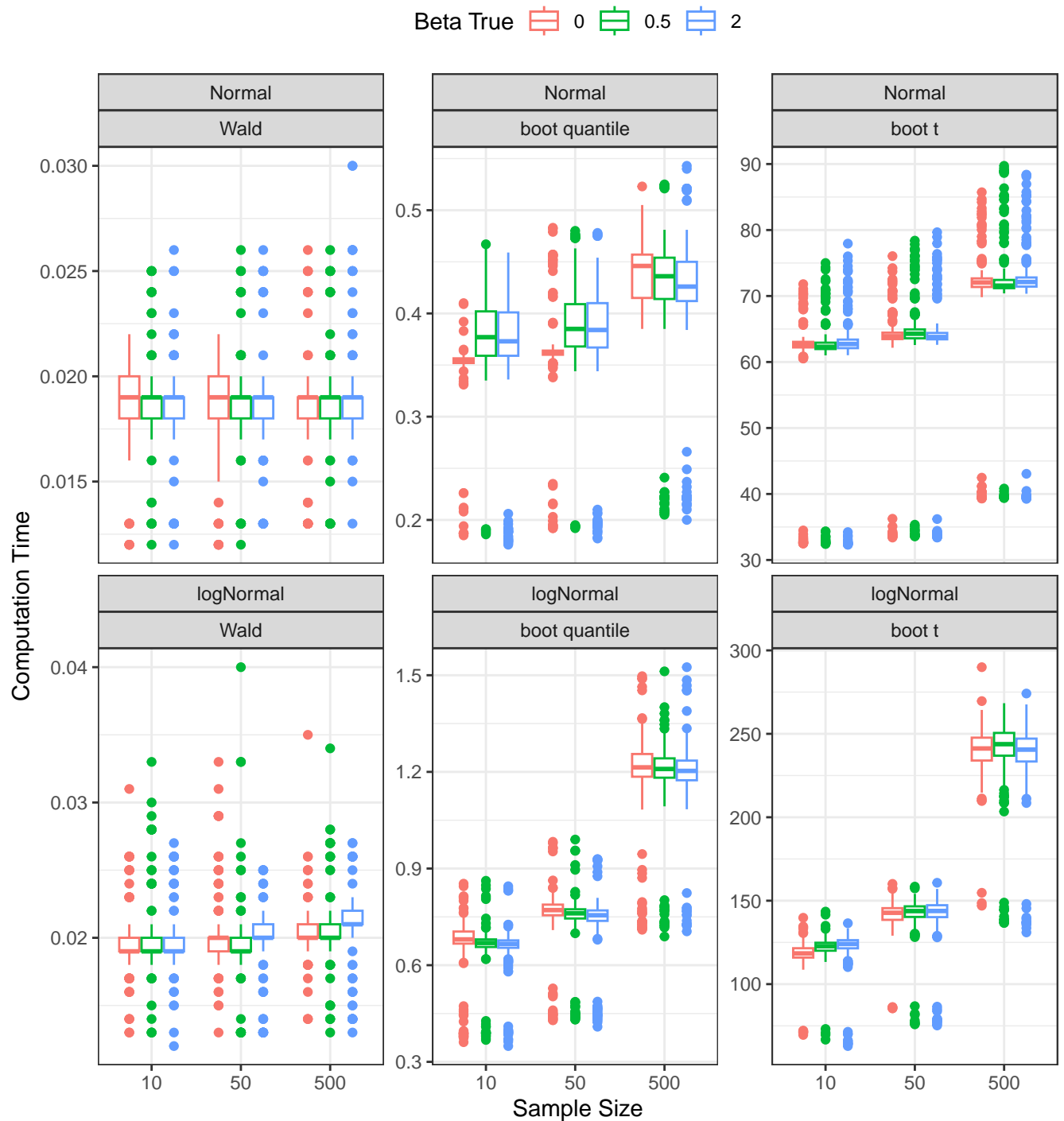
```r
# save plots in results folder
ggsave(plot_bias, file = here::here("results", "plot_bias.jpg"),
       width = 7, height = 6)
ggsave(plot_coverage, file = here::here("results", "plot_coverage.jpg"),
       width = 7, height = 6)
```

```
ggsave(plot_se, file = here::here("results", "plot_se.jpg"),
       width = 6, height = 6)
ggsave(plot_time, file = here::here("results", "plot_time.jpg"),
       width = 7, height = 8)
```

Problem 1.5 Discussion

Interpret the results summarized in Problem 1.4. First, write a **paragraph** summarizing the main findings of your simulation study. Then, answer the specific questions below.

The simulation study evaluated three confidence interval (CI) methods—Wald intervals, bootstrapped quantile intervals, and bootstrapped t intervals—across varying sample sizes, treatment effects, and error distributions. Wald CI was the fastest but performed poorly under skewed errors. Boot quantile CI had moderate computation time, while Boot t CI improved coverage in skewed data but required the most computational resources. As sample size increased, bias decreased under the normal distribution but increased in the lognormal distribution when the treatment effect was nonzero, suggesting that GLM with a log link may not be ideal for estimating $\beta_{treatment}$ in this setting. The estimated standard error decreased as sample size grew, with Boot quantile CI producing the smallest standard errors.

- How do the different methods for constructing confidence intervals compare in terms of computation time?

Wald CI is the fastest, boot quantile CI requires Moderate computation time, and boot t CI is the slowest and requires significantly more computation time than the other two methods.

- Which method(s) for constructing confidence intervals provide the best coverage when $\epsilon_i \sim N(0, 2)$?

Wald CI

- Which method(s) for constructing confidence intervals provide the best coverage when $\epsilon_i \sim logNormal(0, \log(2))$?

boot t CI