Hierarchical Logistic Model for Mulit-Center Clinical Trial

P8160 - Group 3A

Kate Colvin (kac2301), Xuanyu Guo(xg2451), Dang Lin (dl3757), Boxiang Tang (bt2654)

2025-03-03

Introduction and Background

A multicenter clinical trial is a trial that enrolls participants at multiple different medical institutions. While there are a number of distinct advantages to multicenter trials, such as the possibilities for a larger sample size and greater generalizability [Jo, 2020], they are also expensive and more complex to execute and analyze. For instance, different trial sites can have different characteristics, such as patient demographics, treatment practices, or geographic factors, that impact the outcome of trial patients. Such variability between trial sites can complicate the statistical analysis, and possibly bias trial outcomes. Moreover, in some cases, multicenter clinical trials have contradicted single-center trials [Dechartres, 2011]. Therefore an important goal of statistical clinical trial research must be to create methods and guidelines for the reliable design and analysis of multicenter clinical trials.

A challenge in evaluating clinical trial methodology is the cost of conducting a trial and the accessibility of this data to external researchers. It's been long recognized that simulation studies can play a key role in designing and analyzing clinical trials by improving their efficiency and reliability [Bonate, 2000]

In this project, we designed a simulation study to determine optimal Monte Carlo sampling strategies to evaluate the population-level probability of adverse events in the context of a hypothetical multicenter clinical trial. This simulated trial includes patient-level covariates and clinic-level random effects that informs a heirarchical logistic model to predict the probability of adverse events. We evaluated the bias and variance in the predicted probability of an adverse event of three sampling methods: simple Monte Carlo (MC), MC with control variates (CV), and MC with importance sampling (IS). We also compare the CPU time statistics for each method.

Statistical Methods

In order to evaluate our different Monte Carlo sampling stratgies, we created a synthetic data set meant to represent a hypothetical multicenter clinical trial. This data set includes patient-level covariates X_i and clinic-level random effects b_j, which represents site-specific variables such as population demographics and differences in protocol. In section 1 below, we specify the clinic random effect as b_j ~ Log-Normal(-1, 0.5), which is a heavy tailed distribution (i.e. decays slower than an Exponential distribution), and the patient level covariates were specified as X_i ~ Gamma(2, 2). Distribution parameters were chosen to yield realistic estimates of adverse events [Luo, 2016].

These random variables are used to estimate the population-level probability of an adverse event across all patients and clinics using the logit function below:

$$P(AdverseEvent) = \int \int logit^{-1}(\alpha + b + \beta x) f_B(b) f_X(x) db dx$$

We evaluated three different Monte Carlo sampling methods: simple Monte Carlo (MC), MC with control variates (CV), and MC with importance sampling (IS) for evaluating the integral above. The equations for each method are given below, where X_i and b_i are selected from the appropriate probability distributions

$f_X(x)$ and $f_B(b)$, respectively, $U(x_i, b_i)$ is a control variate with a known expectation value, μ_U, and $h(x_i, b_i)$ is a easy to sample bivariate distribution that is non-zero over the same range as $f_X(x_i)f_B(b_i)$.

$$P_{Simple}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N} logit^{-1}(x_i, b_i)$$

$$P_{CV}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N}(logit^{-1}(x_i, b_i) - c^*(U(x_i, b_i)) - \mu_U)$$

$$P_{IS}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N} \frac{f_X(x_i)f_B(b_i)}{h(x_i,b_i)} logit^{-1}(x_i, b_i)$$

For each method, we report the bias and variance in the predicted probability of an adverse event from each method, where the bias is calculated in comparison to the "true probability", which was estimated using a simple Monte Carlo simulation with N = *FINAL N*. We also provide CPU time statistics to gauge the efficiency of each method and cumulative convergence plots to compare the convergence of each method to the "true probability".

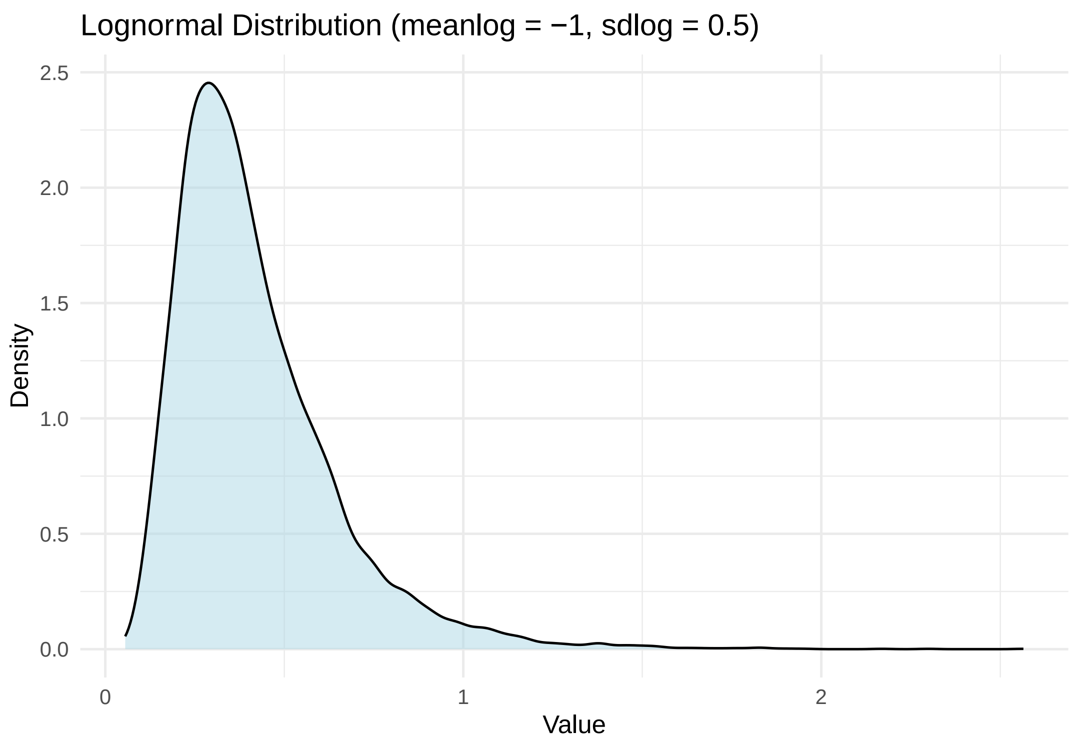
In the sections below, we defined functions used to run the integrations using simple Monte Carlo (Section 2), control variates (Section 3), and importance sampling (Section 4) methods.

Task 1: Specifying Distributions

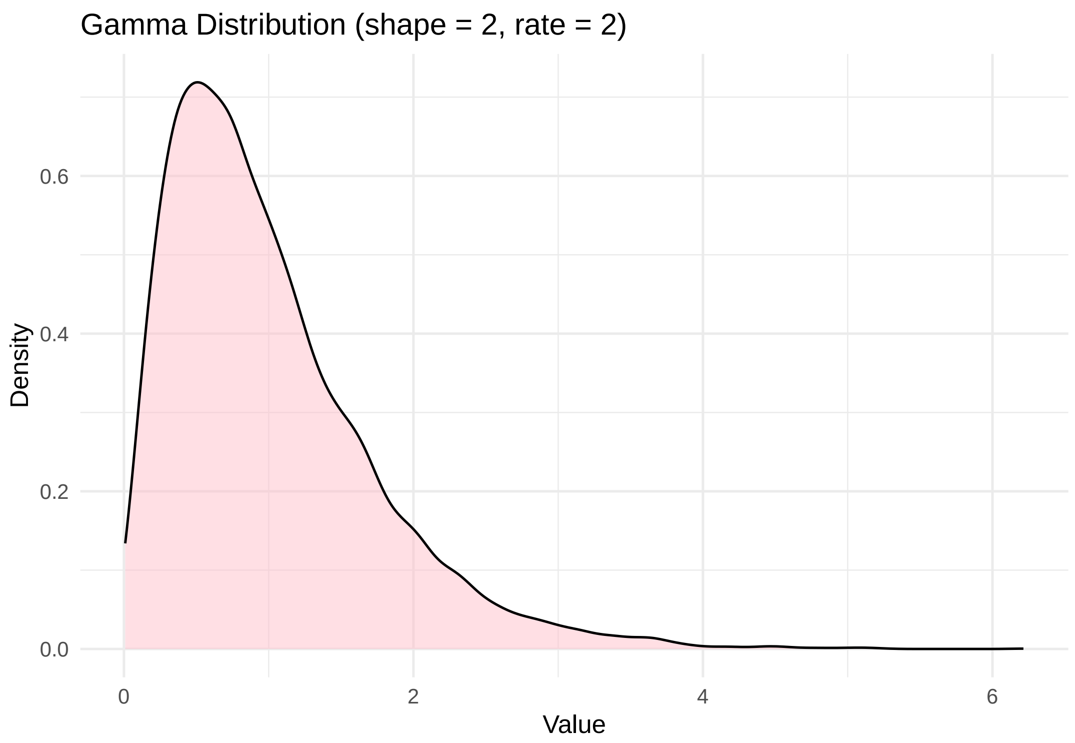
Let b follow a Log-normal distribution with parameters meanlog = -1 and sdlog = 0.5, see graph below.

Let x follows a Gamma distribution with parameters shape = 2 and rate = 2, see graph below.

```r
set.seed(8160)
n <- 10000

## Pre-calculating b samples and x samples for use in simple MC and CV
b_samples <- rlnorm(n, meanlog = -1, sdlog = 0.5)
x_samples <- rgamma(n, shape = 2, rate = 2)

lognormal_df <- data.frame(values = b_samples,
                           distribution = "Lognormal(-1, 0.5)")
gamma_df <- data.frame(values = x_samples,
                       distribution = "Gamma(2, 2)")

ggplot(lognormal_df, aes(x = values)) +
  geom_density(fill = "lightblue", alpha = 0.5) +
  labs(title = "Lognormal Distribution (meanlog = -1, sdlog = 0.5)",
       x = "Value", y = "Density") +
  theme_minimal()
```



```r
ggplot(gamma_df, aes(x = values)) +
  geom_density(fill = "pink", alpha = 0.5) +
  labs(title = "Gamma Distribution (shape = 2, rate = 2)",
       x = "Value", y = "Density") +
  theme_minimal()
```



Task 2: Sampling using Simple Monte Carlo (MC)

Using simple Monte Carlo with N = 100000: $P_{Simple}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N} logit^{-1}(x_i, b_i)$

```r
start_time_smc <- Sys.time()

logit_inverse_function <- function(x) {
  return(exp(x) / (1 + exp(x)))
}

## Calculating the probability of an adverse event based on pre-sampled b and x
p_ij <- logit_inverse_function(alpha + b_samples + beta * x_samples)

p_adverse_event_smc <- mean(p_ij)
var_smc <- var(p_ij)

cpu_time_smc <- Sys.time() - start_time_smc

## knitr::kable(p_adverse_event_smc, digits = 6,
##             col.names = "Probability of the Adverse Event of Simple MC Sampling")
```

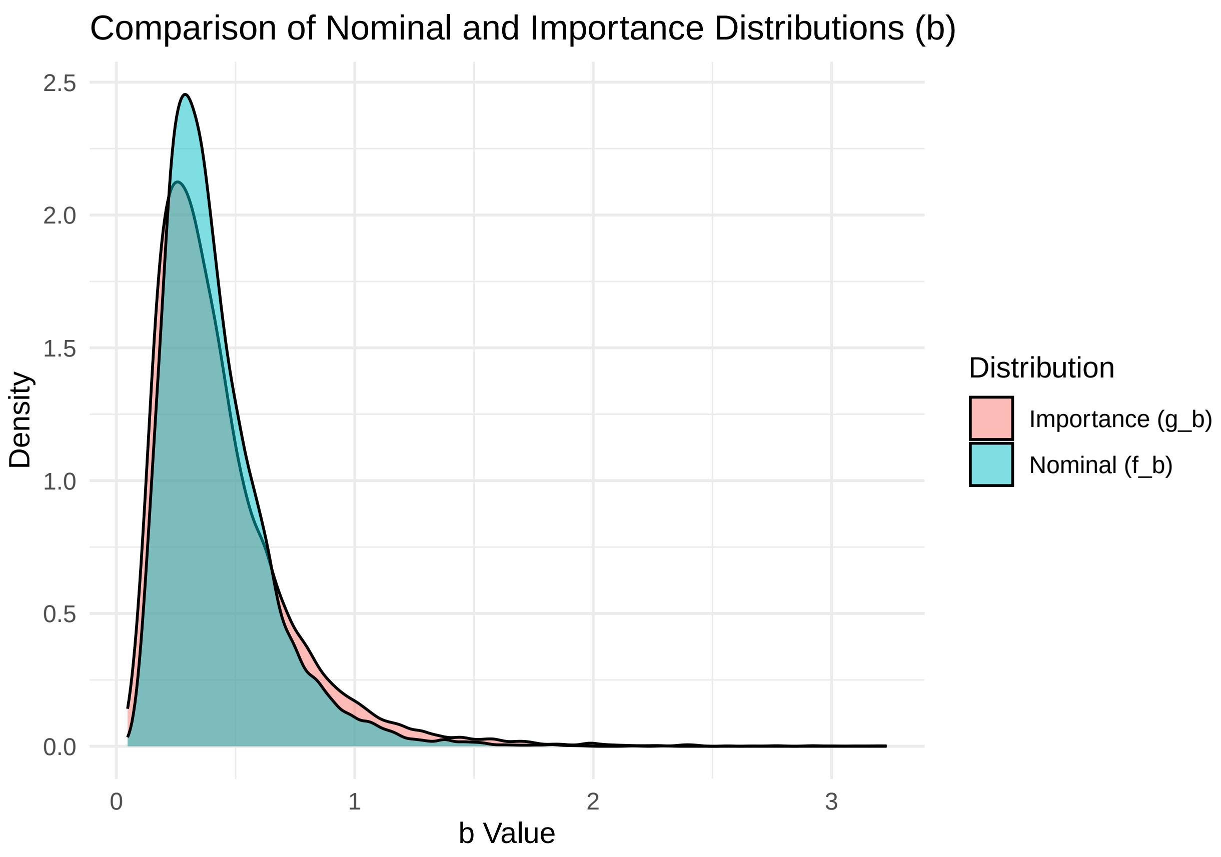
Task 3: Sampling using Control Variate (CV)

Using Monte Carlo with Control Variate: $P_{CV}(AdverseEvent) = \frac{1}{N}\sum_{i=1}^{N}(logit^{-1}(x_i, b_i) - c^*(U(x_i, b_i)) - \mu_U)$, where the control variate (U) is the argument to the inverse logit function, $\alpha + b + \beta x$

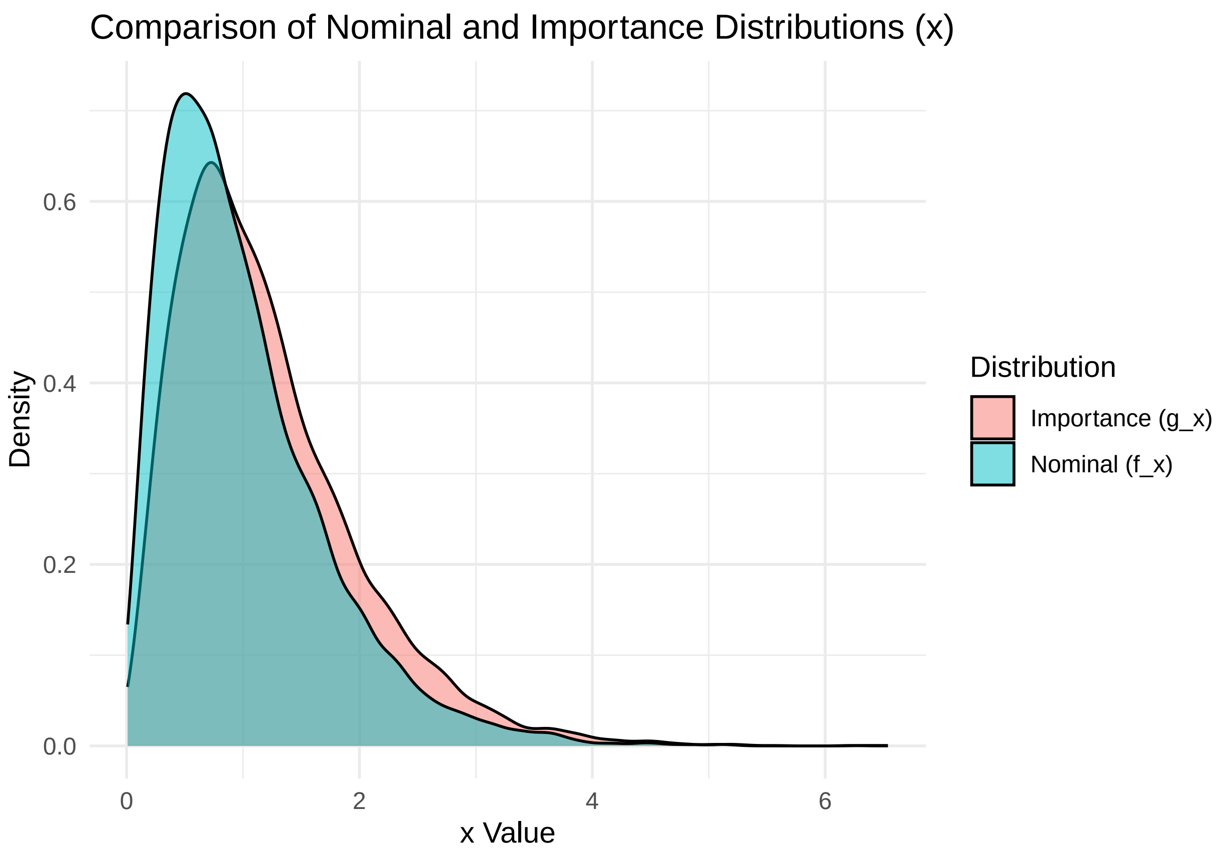
```r
start_time_cv <- Sys.time()

Y <- logit_inverse_function(alpha + b_samples + beta * x_samples)
E_Y <- mean(Y)
var_Y <- var(Y)

## U is the logit transformed Y (i.e. the original argument to the inverse logit)
U <- alpha + b_samples + beta * x_samples
E_U <- mean(U)
var_U <- var(U)

cov_Y_U <- cov(Y, U)
c_star <- -cov_Y_U / var_U

## Applying covariate correction
Y_cv <- Y + c_star * (U - E_U)
E_Y_cv <- mean(Y_cv)
var_Y_cv <- var_Y - (cov_Y_U ^ 2) / var_U

cpu_time_cv <- Sys.time() - start_time_cv

control_variate_df <- data.frame(
  Estimates = c("Mean", "Variance"),
  Without_Control_Variate = c(E_Y, var_Y),
  With_Control_Variate = c(E_Y_cv, var_Y_cv)
)

## knitr::kable(control_variate_df, digits = 10,
##             caption = "Comparison of Estimates With and Without Control Variate")
```

Task 4: Sampling using Importance Sampling (IS)

```r
set.seed(8160)

b_samples_nominal <- rlnorm(n, meanlog = -1, sdlog = 0.5)
x_samples_nominal <- rgamma(n, shape = 2, rate = 2)
b_samples_importance <- rlnorm(n, meanlog = -1, sdlog = 0.6)
x_samples_importance <- rgamma(n, shape = 2.4, rate = 2)

b_df <- data.frame(Value = c(b_samples_nominal, b_samples_importance),
                   Distribution = rep(c("Nominal (f_b)", "Importance (g_b)"), each = n))

x_df <- data.frame(Value = c(x_samples_nominal, x_samples_importance ),
                   Distribution = rep(c("Nominal (f_x)", "Importance (g_x)"), each = n))

ggplot(b_df, aes(x = Value, fill = Distribution)) +
  geom_density(alpha = 0.5) +
  labs(title = "Comparison of Nominal and Importance Distributions (b)",
       x = "b Value", y = "Density") +
  theme_minimal()
```



```r
ggplot(x_df, aes(x = Value, fill = Distribution)) +
  geom_density(alpha = 0.5) +
  labs(title = "Comparison of Nominal and Importance Distributions (x)",
       x = "x Value", y = "Density") +
  theme_minimal()
```



```r
set.seed(8160)

start_time_is <- Sys.time()

# Nominal Distribution
f_b <- function(b) {
  return(dlnorm(b, meanlog = -1, sdlog = 0.5))
  }
f_x <- function(x) {
  return(dgamma(x, shape = 2, rate = 2))
  }

# Importance Distribution
g_b <- function(b) {
  return(dlnorm(b, meanlog = -1, sdlog = 0.6))
}
g_x <- function(x) {
  return(dgamma(x, shape = 2.4, rate = 2))
}

weights <- function(b, x) {
  return((f_b(b) * f_x(x)) / (g_b(b) * g_x(x)))
}

b_samples_importance <- rlnorm(n, meanlog = -1, sdlog = 0.6)
x_samples_importance <- rgamma(n, shape = 2.4, rate = 2)
```

```r
p_ij_is <- logit_inverse_function(alpha + b_samples_importance + beta * x_samples_importance)
w <- weights(b_samples_importance, x_samples_importance)

E_is <- sum(w * p_ij_is) / sum(w)
var_is <- sum(w ^ 2 * (p_ij_is - E_is) ^ 2) / (sum(w) ^ 2)

cpu_time_is <- Sys.time() - start_time_is

importance_sampling_results <- data.frame(
  Estimates = c("Mean", "Variance"),
  Value = c(E_is, var_is)
)

knitr::kable(importance_sampling_results, digits = 6,
             caption = "Importance Sampling Estimates for Adverse Event Probability")
```

Table 1: Importance Sampling Estimates for Adverse Event Probability

Estimates	Value
Mean	0.260451
Variance	0.000001

Results

Task 5: Comparison of Sampling Methods

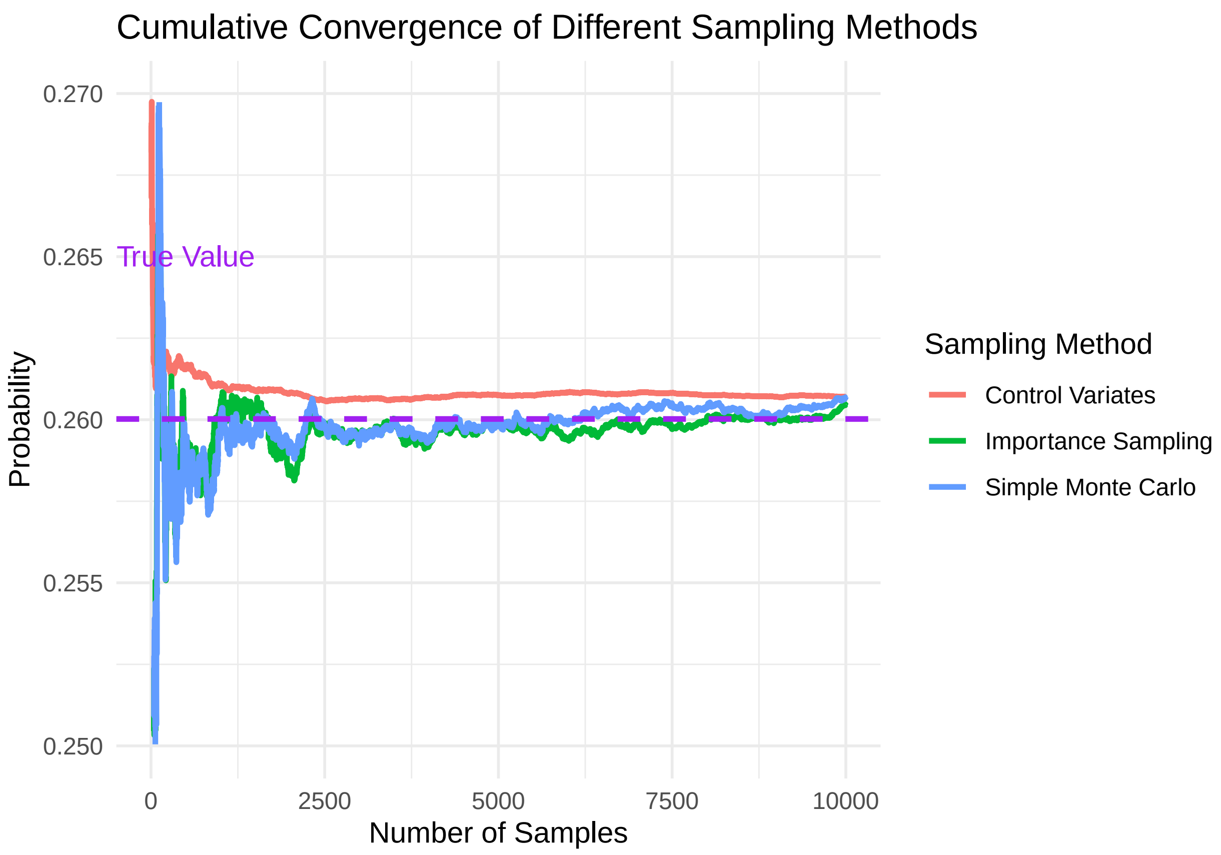
```r
set.seed(8160)

N_true <- 1000000

b_samples_N <- rlnorm(N_true, meanlog = -1, sdlog = 0.5)
x_samples_N <- rgamma(N_true, shape = 2, rate = 2)

p_adverse_event_true <- mean(logit_inverse_function(alpha + b_samples_N + beta * x_samples_N))

bias_smc <- p_adverse_event_smc - p_adverse_event_true
bias_cv <- E_Y_cv - p_adverse_event_true
bias_is <- E_is - p_adverse_event_true

comparison_df <- data.frame(
  Method = c("Simple Monte Carlo", "Control Variates", "Importance Sampling"),
  Estimated_Probability = c(p_adverse_event_smc, E_Y_cv, E_is),
  Bias = c(bias_smc, bias_cv, bias_is),
  Variance = c(var_smc, var_Y_cv, var_is),
  CPU_Time = c(cpu_time_smc, cpu_time_cv, cpu_time_is)
)

knitr::kable(comparison_df, digits = 10,
      caption = "Comparison of Sampling Methods (Bias, Variance, and CPU Time)")
```

Table 2: Comparison of Sampling Methods (Bias, Variance, and CPU Time)

Method	Estimated_Probability	Bias	Variance	CPU_Time
Simple Monte Carlo	0.2606976	0.0006749374	0.0072318122	0.001256943 secs
Control Variates	0.2606976	0.0006749374	0.0000800636	0.003453016 secs
Importance Sampling	0.2604511	0.0004284510	0.0000006625	0.004766941 secs

Task 6: Cumulative Convergence Plots (Extra Credit)

```r
cumulative_mc <- cumsum(p_ij) / seq_along(p_ij)
cumulative_cv <- cumsum(Y_cv) / seq_along(Y_cv)
cumulative_is <- cumsum(w * p_ij_is) / cumsum(w)

cumulative_convergence_df <- data.frame(
  Sample = rep(1 : 10000, 3),
  Value = c(cumulative_mc, cumulative_cv, cumulative_is),
  Method = rep(c("Simple Monte Carlo", "Control Variates", "Importance Sampling"), each = n)
)

ggplot(cumulative_convergence_df, aes(x = Sample, y = Value, color = Method)) +
  geom_line(size = 1) +
  geom_hline(yintercept = p_adverse_event_true, linetype = "dashed", color = "purple", size = 1) +
  annotate("text", x = 500, y = p_adverse_event_true + 0.005,
           label = "True Value", size = 4, color = "purple") +
  labs(title = "Cumulative Convergence of Different Sampling Methods",
       x = "Number of Samples",
       y = "Probability",
       color = "Sampling Method") +
  theme_minimal() +
  ylim(c(0.25, 0.27))
```



Discussion and Practical Implications

The ability to effectively simulate

Conclusion

References

Jo, Daehyun. "The interpretation bias and trap of multicenter clinical research." The Korean journal of pain vol. 33,3 (2020): 199-200. doi:10.3344/kjp.2020.33.3.199

Dechartres, Agnes, et al. "Single-center trials show larger treatment effects than multicenter trials: evidence from a meta-epidemiologic study." Annals of internal medicine 155.1 (2011): 39-51.

Bonate, Peter L. "Clinical trial simulation in drug development." Pharmaceutical research 17 (2000): 252-256.

Luo, Jake, et al. "Population analysis of adverse events in different age groups using big clinical trials data." JMIR medical informatics 4.4 (2016): e6437

3. Cheng, Adam, et al. "Conducting multicenter research in healthcare simulation: Lessons learned from the INSPIRE network." Advances in Simulation 2 (2017): 1-14.