Simulation Study: Optimal Design for Cluster Randomized Trials

Morgan Cunningham

2024-11-27

Abstract

Abstract

This simulation study explores the optimal design of cluster randomized trials under budget constraints. The aim is to evaluate how the number of clusters and patients per cluster influence the estimation of the treatment effect in two scenarios: normal outcomes (continuous, constant variance) and Poisson outcomes (count data with mean-dependent variance). Key factors include variability at the cluster and patient levels (γ^2, σ^2) and the relative costs of recruiting the first and subsequent patients in a cluster (c_1, c_2).

Results highlight that increasing the number of clusters reduces variance more effectively than increasing patients per cluster, emphasizing the importance of prioritizing cluster recruitment within budget constraints. For Poisson outcomes, designs with fewer clusters or patients exhibit higher mean bias and variance due to the hierarchical structure and mean-dependent variability. Cost analyses demonstrate a trade-off between cluster size and patient count, with diminishing returns on precision as budgets increase. These findings highlight the need to balance resource allocation across cluster and patient recruitment to achieve efficient study designs.

The study concludes that designs for normal and Poisson outcomes share similar principles, with clusters playing a critical role in reducing variability and improving precision. Poisson outcomes, however, require additional consideration of their mean-variance relationship and hierarchical dependencies.

A: Aims of the Study

The goal of this simulation study is to evaluate how different combinations of clusters (G) and observations per cluster (R) impact the estimation of the treatment effect beta. We will use hospitals for clusters and patients as the observations per cluster. We ultimately want to optimize study designs under budget constraints, accounting for initial costs c1 for first patient at each hospital and c2 additional patients. We will compare performance metrics such as bias and variance under two distribution settings:

- Normal Outcomes: Continuous outcome with constant variance.
- Poisson Outcomes: Count data with mean-dependent variance.

Lastly, we will explore how variability at the cluster and patient levels, as well as relative costs, affect the optimal study design.

D: Data-Generating Mechanisms

The data-generating process for both distributions are below:

1. Normal Outcomes The hierarchical model for normal outcomes is:

- *Cluster-Level Model:*
$$\mu_i = \alpha + \beta X_i + \epsilon_i, \quad \epsilon_i \sim N(0, \gamma^2)$$
where μ_i is the mean outcome for cluster i, affected by treatment (X_i) and cluster-level variability (γ^2).
- *Observation-Level Model:*
$$Y_{ij}|\mu_i \sim N(\mu_i, \sigma^2)$$
where Y_{ij} is the continuous outcome for observation j in cluster i.

2. Poisson Outcomes The hierarchical model for Poisson outcomes is:

- *Cluster-Level Model:*
$$\log(\mu_i) \sim N(\alpha + \beta X_i, \gamma^2)$$
where μ_i is the log-transformed mean outcome for cluster i.
- *Observation-Level Model:*
$$Y_{ij}|\mu_i \sim \text{Poisson}(\mu_i)$$
where Y_{ij} is the count outcome for observation j in cluster i.

3. Common Parameters

- α: Baseline outcome where we use control group mean for normal and log-scale mean for Poisson.
- β: Treatment effect or the difference in mean or log-mean between groups.
- γ^2: Between-cluster variability.
- σ^2: Within-cluster variability which is only for normal outcomes.

4. Costs

- c_1: Cost of the first patient in a cluster.
- c_2: Cost of subsequent patients, proportional to c_1 ($c_2 = c_1 \times$ cost ratio).

E: Estimands

The treatment effect or β for normal outcomes is the difference in mean outcomes between treatment and control groups. For Poisson outcomes it is the difference in log-mean outcomes between treatment and control groups.

Our performance metrics are as follows:

- *Bias:* Bias $= \hat{\beta} - \beta$.
- *Variance:* Variability in $\hat{\beta}$ across simulations.
- *Standard Deviation:* Square root of variance.
- *Cost Utilization:* Fraction of the budget used for each design.

M: Methods

For the simulation process we will generate data under the hierarchical models for both normal and Poisson. We will vary design parameters (G, R) and data generation parameters (γ^2, σ^2). We will also impose budget constraints and cost ratios to filter feasible designs. For normal outcomes we fit linear mixed-effects models and for poisson outcomes we fit generalized linear mixed-effects models.

We also make sure to exclude designs exceeding the budget. Lastly, we will compute mean bias, variance, and standard deviation of $\hat{\beta}$ across simulations and then identify optimal designs minimizing variance while staying within budget.

P: Performance Measures

Our key performance metrics are as follows:

- *Mean Bias:* Closeness of the estimated treatment effect to the true value.
- *Variance:* Precision of the treatment effect estimate.
- *Standard Deviation:* Root of variance, indicating variability across simulations.
- *Budget Utilization:* Proportion of the budget used for each design.

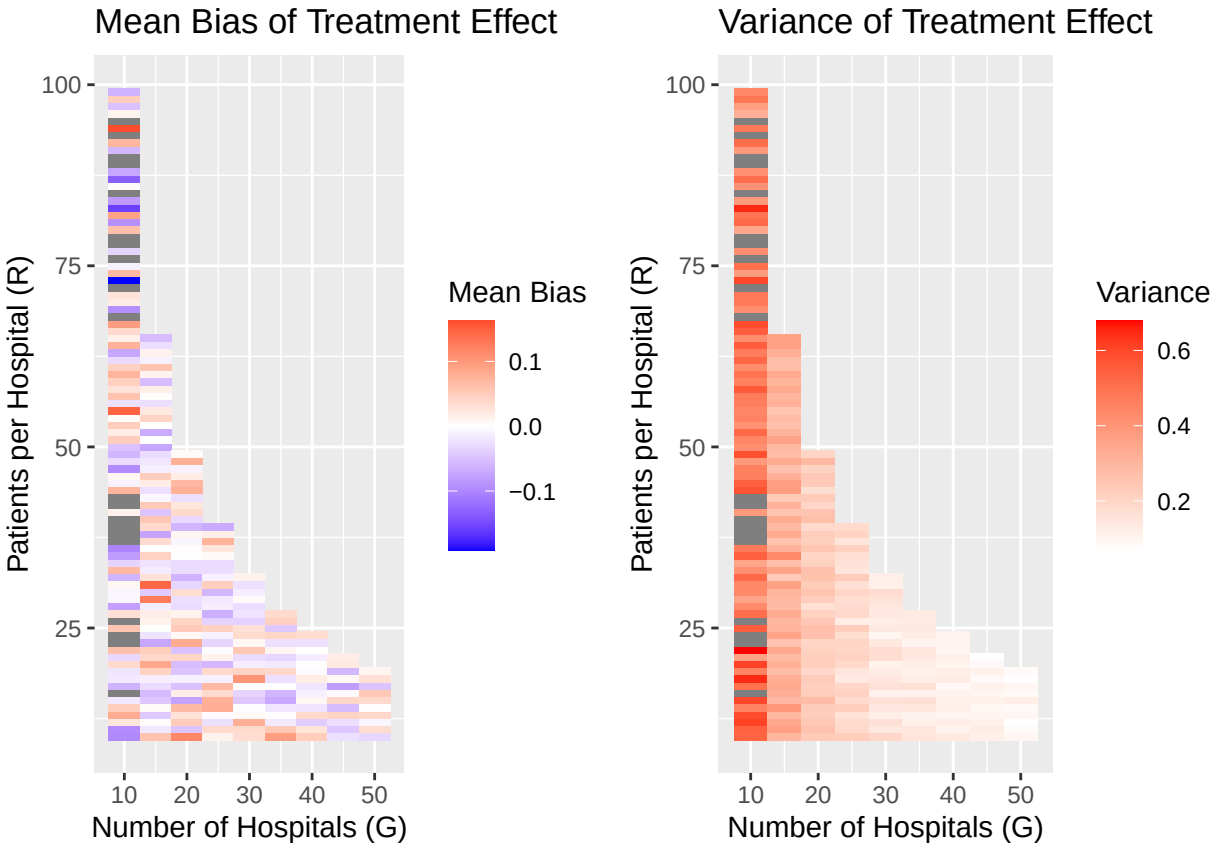
We will also look to compare performance metrics such as the bias and variance for normal and Poisson outcomes across designs.

Normal Outcome

We will first define the parameters for our simulation study designed to evaluate optimal experimental designs for cluster randomized trials under a budget constraint of 5000 units. It specifies that the simulation will run 100 iterations and that the cost of enrolling the first patient in a cluster is 10 units, while the cost for each additional patient in the same cluster is 5 units. The baseline mean outcome for the control group is set to 5, and the treatment effect is defined as 2. Variances at the cluster level and patient level are set at 1 and 2, respectively. The ranges for the number of clusters and the number of patients per cluster are defined as sequences, where the number of clusters ranges from 10 to 50 in increments of 5 and the number of patients per cluster ranges from 10 to 100 in increments of 1.

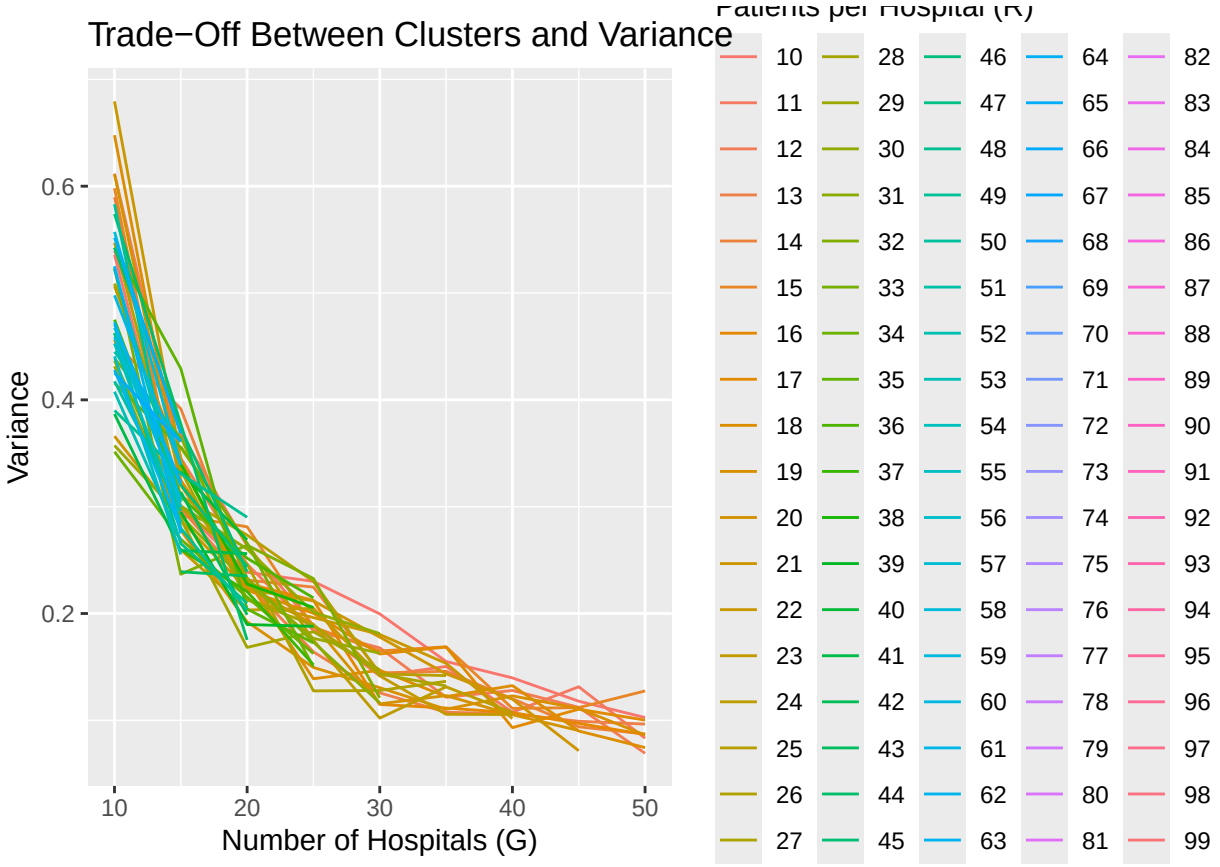
Next, we implement a function, `generate_data`, which simulates hierarchical data based on the specified parameters. The function assigns clusters to either the treatment or control group using a random binomial distribution with a 50% probability for each group. It generates cluster-level random effects from a normal distribution with a mean of 0 and a standard deviation of 1. For each cluster, patient-level outcomes are simulated using a normal distribution with a mean defined as `mu_i = alpha + beta * X[i] + cluster_effects[i]` and a standard deviation of `sqrt(sigma2)`. The function produces a data frame containing the simulated cluster IDs, treatment assignments, and patient-level outcomes. This dataset will be used in further analysis to assess the performance of various study designs under these hierarchical settings.

Next we will run simulation for different combinations of G (clusters/hospitals) and R (patients). We will do this by looping over different cluster sizes and patient numbers, calculating the total cost for each combination. If the cost is within the specified budget, the function runs multiple simulations, generating data and fitting a mixed-effects model to estimate the treatment effect. It then calculates the bias and variance for each simulation by comparing the estimated treatment effect to the true value. Finally, we compile the results, including the mean bias and variance for each combination of cluster size and patient number, and return them in a data frame.



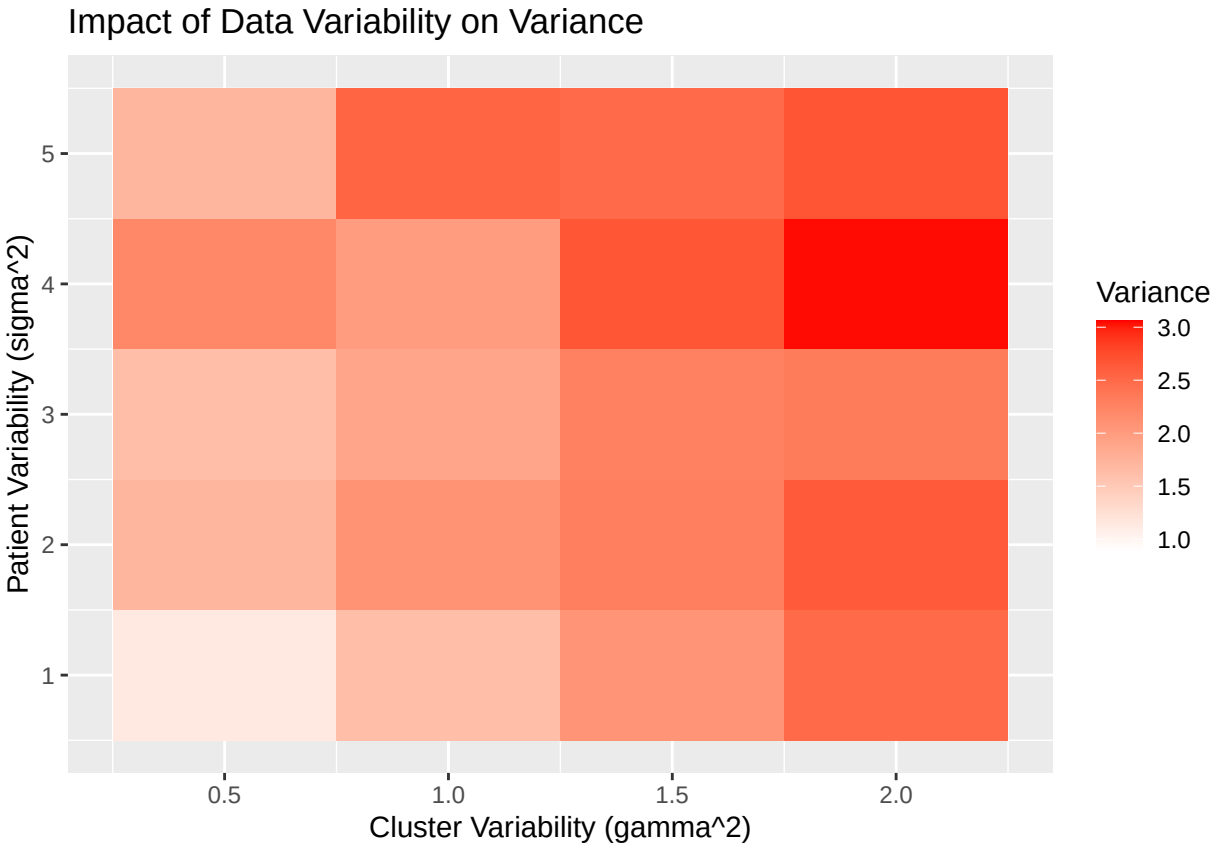
From the first figure, we see that as G increases and there are more clusters and the bias slowly approaches zero, suggesting more accurate estimation with a greater number of clusters and lower number of patients per cluster. At lower values of G, bias is more pronounced, likely because fewer clusters lead to higher susceptibility to variability in cluster-level effects. Increasing R seems to have less impact on reducing bias compared to increasing G.

In the variance of treatment effect figure, we see variance decreases significantly as G increases, indicating that more clusters lead to more precises estimates of the treatment effect. Variance also decreases moderately as R increases, although this effect is still less pronounced than for G. We see an obvious trade-off between G and R, that being, increasing G reduces variance more effectively than increasing R given the same budget. This in itself suggests that allocating resources to recruit more clusters rather than more observations per cluster is generally better for reducing variance.

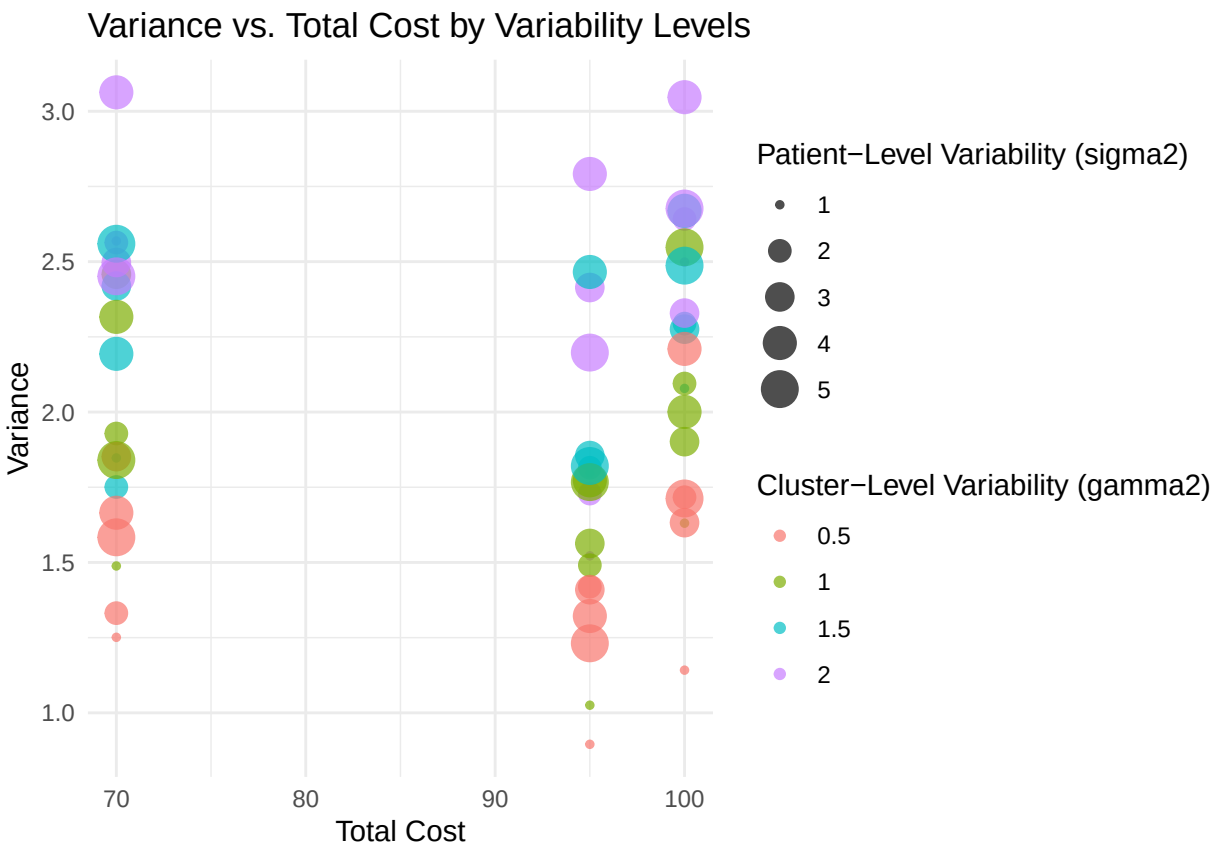


We look a little bit closer at how the number of clusters and patients per cluster impacts the variance of the treatment effect estimate. Variance decreases sharply as G increases, particularly at lower values, emphasizing that adding more clusters significantly improves precision. However, beyond a certain point ($G > 30$), the rate of reduction flattens, indicating diminishing returns. The effect of R, represented by color-coded lines, is relatively minor compared to G. While increasing R slightly reduces variance, its impact is limited, suggesting that precision is driven more by the number of clusters than by the number of patients per cluster. The figure also reveals a balance between G and R. Adding clusters is more effective for reducing variance, but sufficient patients per cluster are still necessary to capture within-cluster variability.

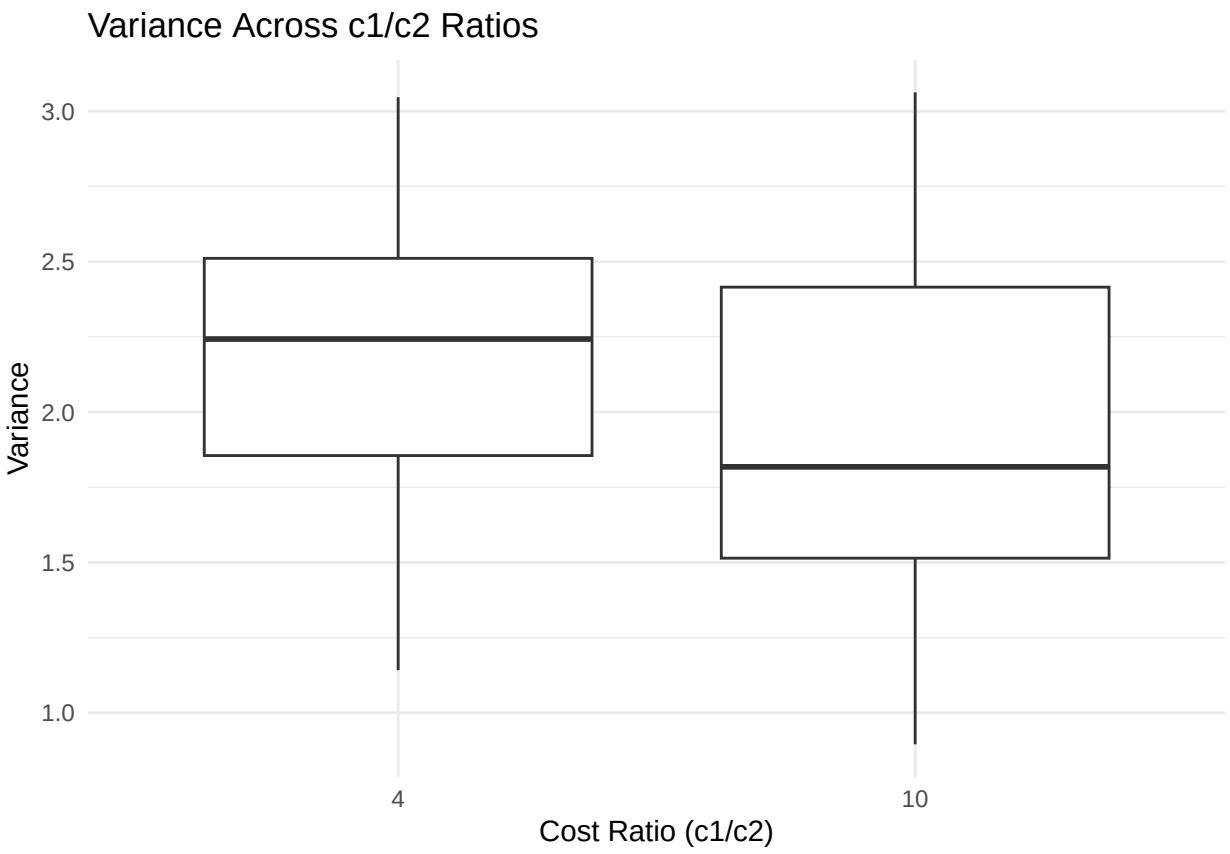
We will now explore how cluster-level variability, patient-level variability, and relative costs affect the variance of the treatment effect estimate and the optimal design of a cluster randomized trial. Previously, we examined how varying the number of clusters and patients per cluster impacts bias and variance in the treatment effect estimate within a fixed budget. This next step extends the analysis by systematically varying key parameters of the data-generating mechanism and cost structure to explore their combined influence on study design. The figures and results we looked at earlier were limited to fixed variability and cost settings. This code introduces flexibility, enabling us to assess how different settings for variability and costs influence outcomes.



We look at the combined effect of cluster-level variability and patient-level variability on the variance of the treatment effect estimate. Variance increases with higher values of both sigma squared and gamma squared, indicating that variability at both levels contributes to reduced precision in estimating the treatment effect. However, the impact of gamma squared appears more pronounced at higher levels, as evidenced by the darker red hues in the upper-right region of the plot. This suggests that designs aimed at minimizing cluster-level variability, such as through careful randomization or stratification, can substantially reduce variance.



Building on the insights from the first figure, this scatterplot demonstrates how variance interacts with total cost while accounting for both cluster-level and patient-level variability. Larger dots represent higher sigma squared, while colors indicate different levels of gamma squared. Higher total costs generally correspond to slightly lower variance, but this effect diminishes as both gamma squared and sigma squared increase. The trend highlights that higher variability levels (larger dots and darker hues) constrain the benefits of increased budget allocations, reinforcing the importance of addressing variability in study design to maximize the efficiency of resource use.



This boxplot focuses on how the relative costs of the first patient and subsequent patients within a cluster affect variance. Higher c1/c2 ratios (10) are associated with slightly lower variances compared to lower ratios (4), as evidenced by the box positions. This indicates that designs where the cost of additional patients is relatively lower lead to greater within-cluster variability, potentially due to increased patient-level contributions that amplify the effects of. These findings suggest that balancing the allocation of budget between cluster recruitment and patient recruitment is critical for optimizing variance reduction.

Poisson Outcome

Next we will explore how hierarchical count data impacts design decisions. Unlike normal outcomes, where Y is continuous and follows a Gaussian distribution, Poisson outcomes are count-based and exhibit mean-dependent variance. This difference requires a hierarchical model tailored to the Poisson distribution, which accommodates the log-linear relationship between covariates and the mean outcome. It can also amplify the effects of cluster-level variability and skew design trade-offs. By generating performance metrics for various combinations of G and R under realistic cost constraints, the simulation provides interesting insights into optimizing cluster randomized trials for count-based outcomes.

The `generate_data_poisson` function simulates hierarchical data based on the Poisson model. Cluster-level treatment assignments are generated as a binary indicator (control vs. treatment) with equal probability. Random effects at the cluster level are drawn from a normal distribution to capture variability between clusters. For each cluster, the mean outcome on the log scale is determined by the fixed effects and the cluster-level random effects. The log-scale mean is exponentiated to calculate the actual mean, and individual patient outcomes are generated from a Poisson distribution with mean. The function returns a data frame containing cluster IDs, patient IDs, treatment assignments, and outcomes, effectively simulating count data with hierarchical dependencies.

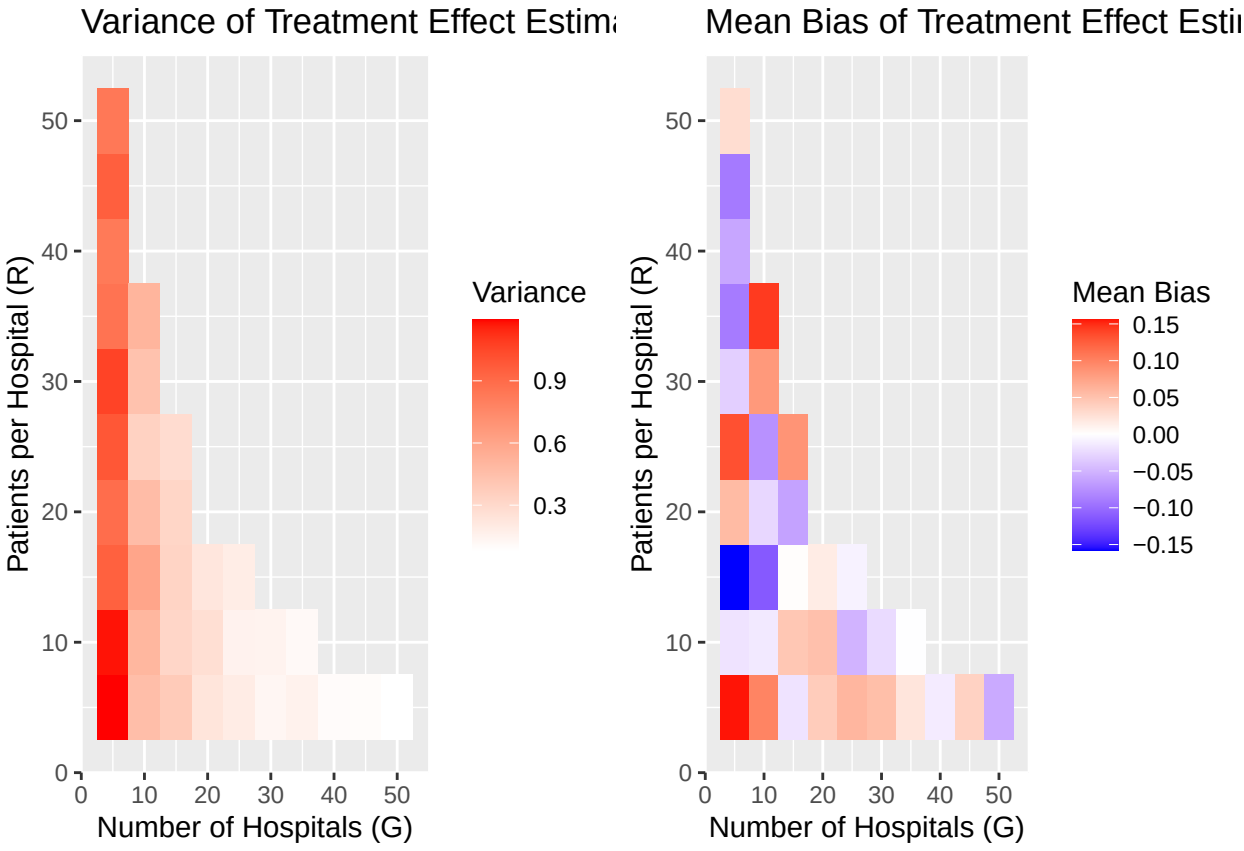
The `simulate_study_poisson` function extends the simulation framework to accommodate Poisson-distributed outcomes. It iterates over combinations of clusters and patients per cluster, constrained by a fixed budget, to evaluate the bias and variance of the treatment effect estimate:

- Cost Calculation: The total cost is calculated based on the cost of the first patient in each cluster and cost of additional patients. Only feasible combinations of G and R within the budget are evaluated.
- Data Simulation: For each feasible combination, hierarchical Poisson data are generated using `generate_data_poisson`.
- Model Fitting: A generalized linear mixed-effects model with a Poisson link function is fitted to the data. The fixed effect estimate for the treatment variable is extracted, handling errors gracefully.
- Performance Metrics: Across multiple simulations, the mean bias and variance of the treatment effect estimates are computed and stored in the results.

Lastly, we execute the simulation across various combinations of G and R. We will fix parameters like α, β, and γ^2 while varying the number of clusters and patients per cluster. The results are stored in `simulation_results_poisson`.

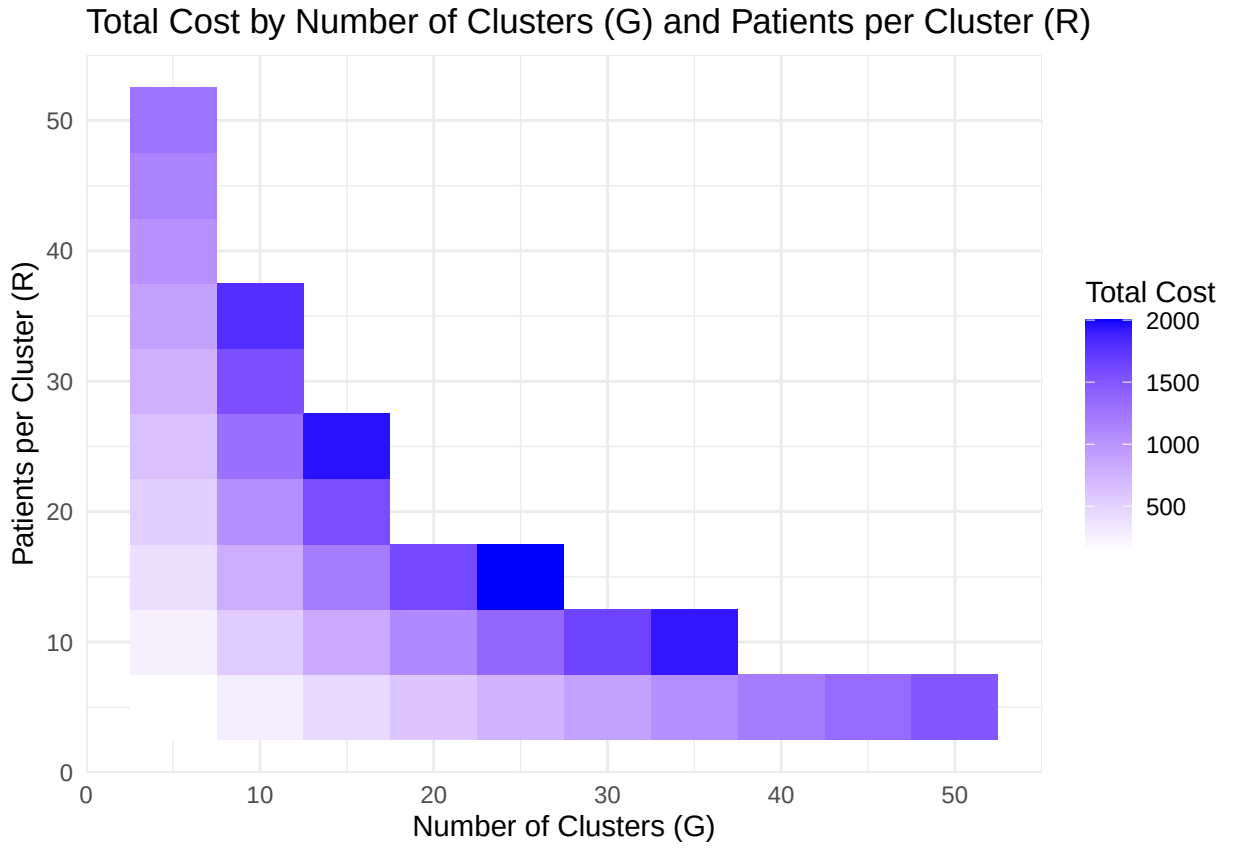
Modify simulation function to work with the Poisson model

Run the simulation for Poisson outcomes across various combinations of G and R

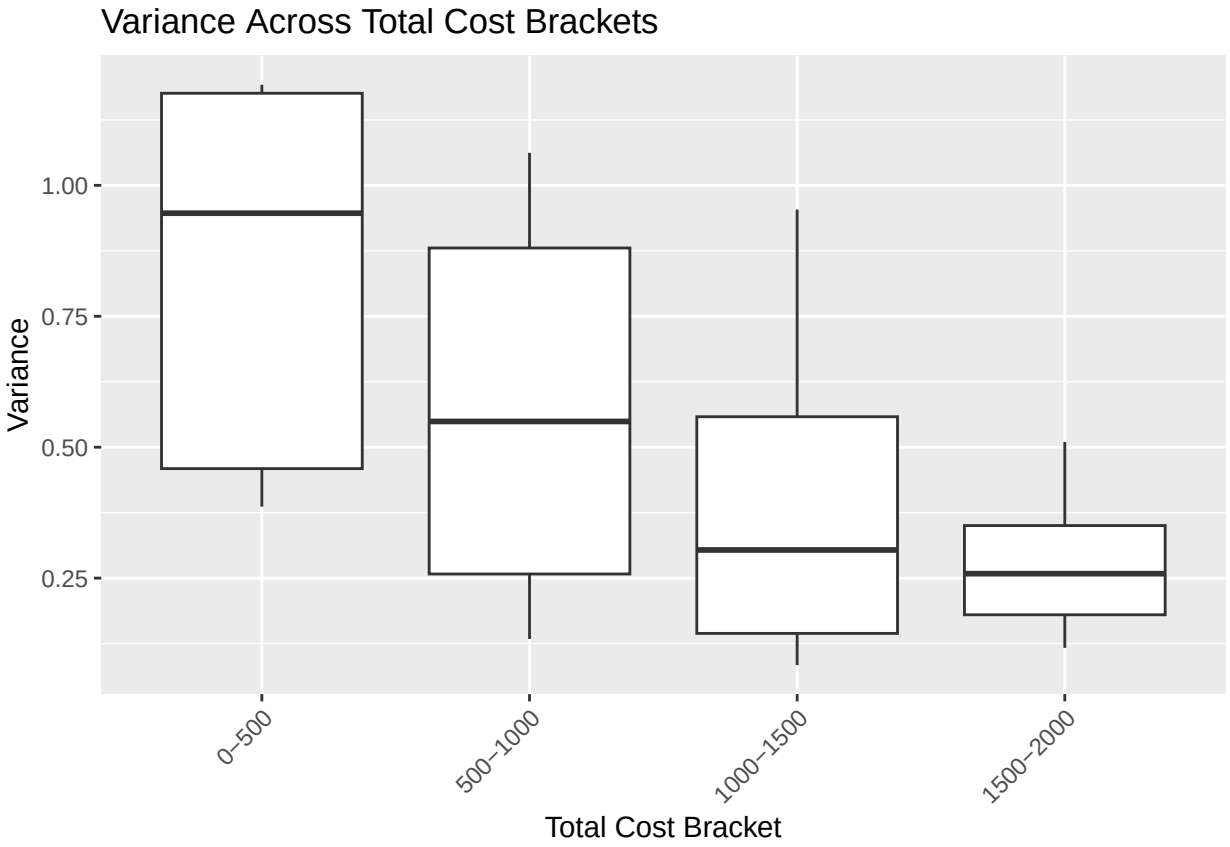


Between the combined heatmaps, we see how variance and mean bias vary with the number of clusters and patients per cluster. Similar to our results for the normal outcome, we see variance decreases as G increases, reflecting the greater precision achieved with more clusters. Conversely, increasing R has diminishing returns

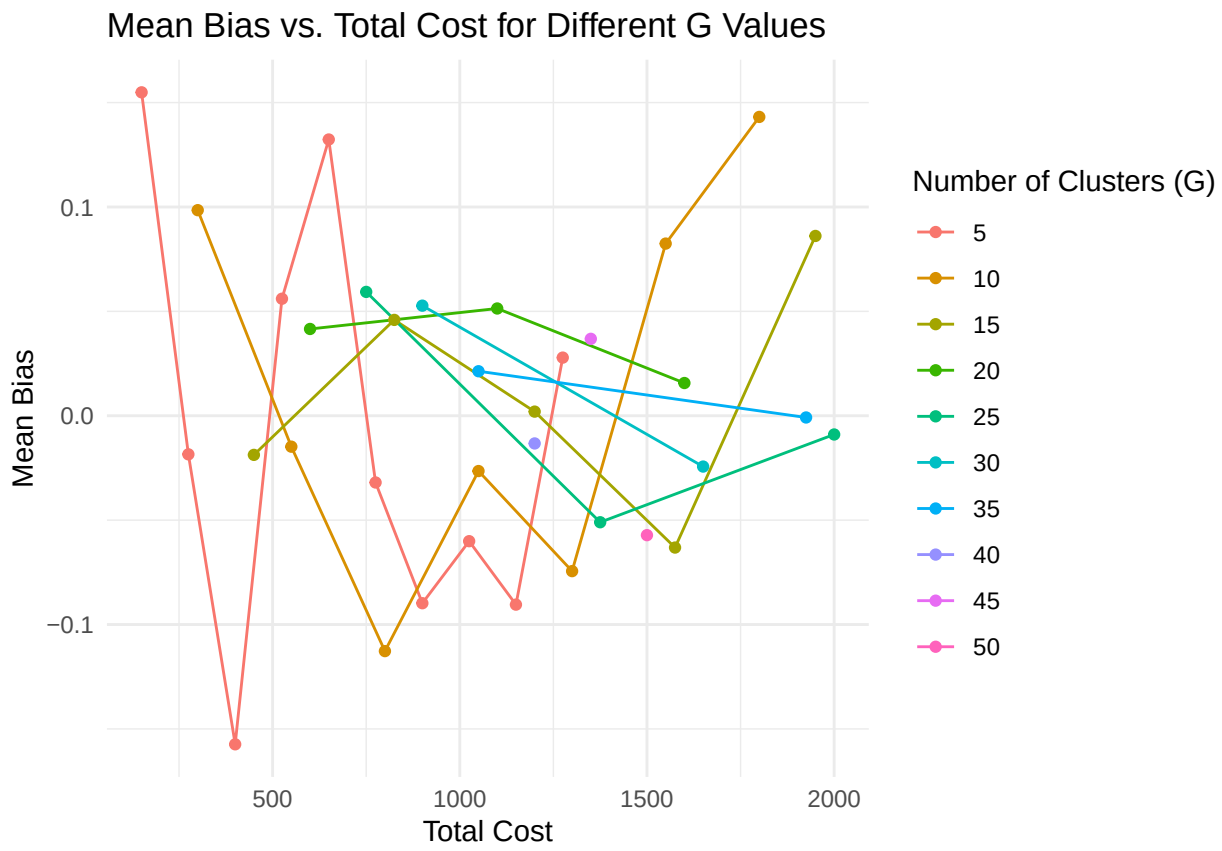
on variance reduction. The mean bias plot shows that bias is most pronounced for smaller values of G and R, indicating underpowered designs. Similar to the normal outcome, these patterns highlight that Poisson outcomes benefit most from designs emphasizing more clusters rather than increasing the number of patients per cluster.



This figure visualizes the cost implications of different combinations of G and R. Designs with high R and G quickly exceed budget constraints, while smaller designs remain feasible within the budget. The gradient emphasizes the trade-off between cluster size and patient count, reinforcing the need for careful cost management in Poisson-distributed trials, where additional patients contribute less to variance reduction than additional clusters.



The boxplot shows a clear trend: variance decreases as total cost increases, but the rate of improvement slows at higher cost brackets. This result indicates that higher budgets allow for better precision in estimating treatment effects which is expected, but the benefit diminishes beyond a certain threshold. The findings align with Poisson outcomes' sensitivity to both cluster-level and patient-level variability, suggesting that resource allocation should prioritize variance reduction efficiently.



Lastly, the line plot explores how mean bias changes with total cost for different cluster counts. Bias generally decreases as total cost increases, particularly for larger G designs. However, the trend is inconsistent at lower G, reflecting the challenges of achieving unbiased estimates in smaller cluster designs. This pattern shows the importance of sufficiently large G to reduce bias in Poisson models, where hierarchical random effects heavily influence outcomes.

Conclusion

This simulation study demonstrates that optimal cluster randomized trial designs depend on careful balancing of cluster and patient recruitment while considering budget constraints and outcome distributions. For both normal and Poisson outcomes, increasing the number of clusters is the most effective strategy for reducing variance and improving the precision of the treatment effect estimate. Poisson outcomes introduce additional challenges, with higher sensitivity to variability and cost structures. Effective designs minimize cluster-level variability while allocating resources toward recruiting more clusters rather than additional patients per cluster.

Code Appendix

```
knitr::opts_chunk$set(message=FALSE,
                      warning=FALSE,
                      error=FALSE,
                      echo = FALSE,
                      fig.pos = "H",
```

```
                      fig.align = 'center')
# Import libraries
library(lme4)
library(ggplot2)
library(dplyr)
library(gridExtra)
library(tidyr)
library(dplyr)
# Simulation
set.seed(123)

n_sim <- 100  # number of simulations
B <- 5000  # total budget
c1 <- 10  # cost of first patient in a hospital
c2 <- 5   # cost of additional patients in same hospital

alpha <- 5  # mean outcome for control group
beta <- 2   # treatment effect
gamma2 <- 1  # variance of hospital-level random effects
sigma2 <- 2  # variance of patient-level random effects

# ranges for clusters (G) and patients per cluster (R)
clusters <- seq(10, 50, by = 5)  # number of hospitals
patients <- seq(10, 100, by = 1)  # patients per hospital
generate_data <- function(G, R, alpha, beta, gamma2, sigma2) {
  # generate cluster-level treatment assignments
  X <- rbinom(G, 1, 0.5)  # 50% hospitals in treatment, 50% in control

  # generate cluster-level random effects
  cluster_effects <- rnorm(G, mean = 0, sd = sqrt(gamma2))

  # generate patient-level outcomes
  data <- data.frame()
  for (i in 1:G) {
    mu_i <- alpha + beta * X[i] + cluster_effects[i]
    patient_outcomes <- rnorm(R, mean = mu_i, sd = sqrt(sigma2))
    cluster_data <- data.frame(
      cluster_id = i,
      patient_id = 1:R,
      treatment = X[i],
      outcome = patient_outcomes
    )
    data <- rbind(data, cluster_data)
  }

  return(data)
}
# fixed simulation
simulate_study <- function(n_sim, clusters, patients, B, c1, c2, alpha, beta, gamma2, sigma2) {
  results <- data.frame()

  for (G in clusters) {
    for (R in patients) {
```

```
      total_cost <- G * c1 + G * (R - 1) * c2  # calculate total cost
      if (total_cost <= B) {  # only run if within budget
        bias <- numeric(n_sim)
        mse <- numeric(n_sim)

        for (sim in 1:n_sim) {
          # generate data
          data <- generate_data(G, R, alpha, beta, gamma2, sigma2)

          # fit mixed-effects model
          #model <- lm(outcome ~ treatment, data = data)
          model <- lmer(outcome ~ treatment + (1 | cluster_id), data = data)

          # extract estimated treatment effect
          est_beta <- fixef(model)["treatment"]

          # performance metrics
          bias[sim] <- est_beta - beta
          mse[sim] <- (est_beta - beta)^2
        }

        # results
        results <- rbind(results, data.frame(
          G = G,
          R = R,
          total_cost = total_cost,
          mean_bias = mean(bias),
          mse = mean(mse)
        ))
      }
    }
  }

  return(results)
}
simulation_results <- simulate_study(
  n_sim = n_sim,
  clusters = clusters,
  patients = patients,
  B = B,
  c1 = c1,
  c2 = c2,
  alpha = alpha,
  beta = beta,
  gamma2 = gamma2,
  sigma2 = sigma2
)
#write.csv(simulation_results, "~/Desktop/PHP 2550 Practical Data Analysis/simFixedResults.csv",row.nam
sims_df <- read.csv("simFixedResults.csv")
# mean bias vs. number of clusters and patients
plot1 <- ggplot(sims_df, aes(x = G, y = R, fill = mean_bias)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
```

```
  labs(title = "Mean Bias of Treatment Effect",
       x = "Number of Hospitals (G)",
       y = "Patients per Hospital (R)",
       fill = "Mean Bias")

# mean squared error (MSE) vs. number of clusters and patients
plot2 <- ggplot(sims_df, aes(x = G, y = R, fill = mse)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Variance of Treatment Effect",
       x = "Number of Hospitals (G)",
       y = "Patients per Hospital (R)",
       fill = "Variance")

grid.arrange(plot1, plot2, ncol=2)
ggplot(sims_df, aes(x = G, y = mse, color = as.factor(R))) +
  geom_line() +
  labs(title = "Trade-Off Between Clusters and Variance",
       x = "Number of Hospitals (G)",
       y = "Variance",
       color = "Patients per Hospital (R)")

# ranges for data generation parameters and costs
gamma2_values <- seq(0.5, 2, by = 0.5)  # cluster-level variability
sigma2_values <- seq(1, 5, by = 1)      # patient-level variability
c1_values <- c(10, 20, 50)              # first patient cost
c2_ratios <- c(0.1, 0.25, 0.5, 0.75, 1) # relative cost ratios (c2 as a proportion of c1)

# expand grid of parameter combinations
parameter_grid <- expand.grid(
  gamma2 = gamma2_values,
  sigma2 = sigma2_values,
  c1 = c1_values,
  c2_ratio = c2_ratios
)
# c2 proportion of c1
parameter_grid$c2 <- parameter_grid$c1 * parameter_grid$c2_ratio
simulate_study_aim2 <- function(n_sim, clusters, patients, B, alpha, beta, parameter_grid) {
  results <- data.frame()

  for (params in 1:nrow(parameter_grid)) {
    gamma2 <- parameter_grid$gamma2[params]
    sigma2 <- parameter_grid$sigma2[params]
    c1 <- parameter_grid$c1[params]
    c2 <- parameter_grid$c2[params]

    for (G in clusters) {
      for (R in patients) {
        total_cost <- G * c1 + G * (R - 1) * c2
        if (total_cost <= B) {
          estimates <- numeric(n_sim)

          for (sim in 1:n_sim) {
```

```
            # generate data
            data <- generate_data(G, R, alpha, beta, gamma2, sigma2)

            # fit model with error handling
            model <- tryCatch(lmer(outcome ~ treatment + (1 | cluster_id), data = data),
                              error = function(e) NA)

            if (!is.na(model) && !isSingular(model)) {
              estimates[sim] <- fixef(model)["treatment"]
            }
          }

          # results
          results <- rbind(results, data.frame(
            gamma2 = gamma2,
            sigma2 = sigma2,
            c1 = c1,
            c2 = c2,
            G = G,
            R = R,
            total_cost = total_cost,
            variance = var(estimates, na.rm = TRUE),
            std_dev = sd(estimates, na.rm = TRUE)
          ))
        }
      }
    }
  }

  return(results)
}
# design space for G and R
clusters <- seq(5, 50, by = 5)  # number of hospitals
patients <- seq(5, 50, by = 5)  # patients per hospital

# simulation
simulation_results2 <- simulate_study_aim2(
  n_sim = 100,
  clusters = clusters,
  patients = patients,
  B = 1000,  # budget
  alpha = 5,  # baseline outcome
  beta = 2,   # true treatment effect
  parameter_grid = parameter_grid
)
#write.csv(simulation_results2, "~/Desktop/PHP 2550 Practical Data Analysis/simMixedResults.csv",row.na
simulation_results2 <- read.csv("simMixedResults.csv")
# variability
 ggplot(simulation_results2, aes(x = gamma2, y = sigma2, fill = variance)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Impact of Data Variability on Variance",
       x = "Cluster Variability (gamma^2)",
```

```
       y = "Patient Variability (sigma^2)",
       fill = "Variance")
ggplot(simulation_results2, aes(x = total_cost, y = variance, color = as.factor(gamma2), size = sigma2)
  geom_point(alpha = 0.7) +
  labs(title = "Variance vs. Total Cost by Variability Levels",
       x = "Total Cost",
       y = "Variance",
       color = "Cluster-Level Variability (gamma2)",
       size = "Patient-Level Variability (sigma2)") +
  theme_minimal()

simulation_results2$c1_c2_ratio <- simulation_results2$c1 / simulation_results2$c2
ggplot(simulation_results2, aes(x = as.factor(c1_c2_ratio), y = variance)) +
  geom_boxplot() +
  labs(title = "Variance Across c1/c2 Ratios",
       x = "Cost Ratio (c1/c2)",
       y = "Variance") +
  theme_minimal()
generate_data_poisson <- function(G, R, alpha, beta, gamma2) {
  # cluster-level treatment assignments
  X <- rbinom(G, 1, 0.5)  # 50% hospitals in treatment, 50% in control

  # cluster-level random effects
  cluster_effects <- rnorm(G, mean = 0, sd = sqrt(gamma2))

  # patient-level outcomes
  data <- data.frame()
  for (i in 1:G) {
    log_mu_i <- alpha + beta * X[i] + cluster_effects[i]
    mu_i <- exp(log_mu_i)  # convert log-scale to mean scale
    patient_outcomes <- rpois(R, lambda = mu_i)  # poisson outcomes
    cluster_data <- data.frame(
      cluster_id = i,
      patient_id = 1:R,
      treatment = X[i],
      outcome = patient_outcomes
    )
    data <- rbind(data, cluster_data)
  }

  return(data)
}

simulate_study_poisson <- function(n_sim, clusters, patients, B, alpha, beta, gamma2, c1, c2) {
  results <- data.frame()

  for (G in clusters) {
    for (R in patients) {
      total_cost <- G * c1 + G * (R - 1) * c2
      if (total_cost <= B) {
        estimates <- numeric(n_sim)

        for (sim in 1:n_sim) {
```

```
          # generate data
          data <- generate_data_poisson(G, R, alpha, beta, gamma2)

          # fit GLMM model w/ Poisson distribution
          model <- tryCatch(glmer(outcome ~ treatment + (1 | cluster_id),
                                  data = data,
                                  family = poisson(link = "log")),
                            error = function(e) NA)

          if (!is.na(model)) {
            estimates[sim] <- fixef(model)["treatment"]
          }
        }

        # results
        results <- rbind(results, data.frame(
          G = G,
          R = R,
          total_cost = total_cost,
          mean_bias = mean(estimates - beta, na.rm = TRUE),
          variance = var(estimates, na.rm = TRUE)
        ))
      }
    }
  }

  return(results)
}

# simulation parameters
clusters <- seq(5, 50, by = 5)  # number of hospitals
patients <- seq(5, 50, by = 5)  # patients per hospital
B <- 2000  # fixed budget
alpha <- 1  # log-scale baseline count
beta <- 0.5  # log-scale treatment effect
gamma2 <- 1  # cluster-level variability
c1 <- 10  # cost of first patient
c2 <- 5   # cost of additional patients

# run simulation
simulation_results_poisson <- simulate_study_poisson(
  n_sim = 100,
  clusters = clusters,
  patients = patients,
  B = B,
  alpha = alpha,
  beta = beta,
  gamma2 = gamma2,
  c1 = c1,
  c2 = c2
)

#write.csv(simulation_results_poisson, "~/Desktop/PHP 2550 Practical Data Analysis/simPoissonResults.cs
```

```r
simulation_results_poisson <- read.csv("simPoissonResults.csv")
plot3 <- ggplot(simulation_results_poisson, aes(x = G, y = R, fill = variance)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Variance of Treatment Effect Estimate (Poisson)",
       x = "Number of Hospitals (G)",
       y = "Patients per Hospital (R)",
       fill = "Variance")

plot4 <- ggplot(simulation_results_poisson, aes(x = G, y = R, fill = mean_bias)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  labs(title = "Mean Bias of Treatment Effect Estimate (Poisson)",
       x = "Number of Hospitals (G)",
       y = "Patients per Hospital (R)",
       fill = "Mean Bias")

grid.arrange(plot3, plot4, ncol=2)
heatmap_data <- aggregate(total_cost ~ G + R, data = simulation_results_poisson, mean)
ggplot(heatmap_data, aes(x = G, y = R, fill = total_cost)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Total Cost by Number of Clusters (G) and Patients per Cluster (R)",
       x = "Number of Clusters (G)",
       y = "Patients per Cluster (R)",
       fill = "Total Cost") +
  theme_minimal()

simulation_results_poisson$total_cost_bracket <- cut(simulation_results_poisson$total_cost,
                                                     breaks = seq(0, max(simulation_results_poisson$tota
                                                     labels = paste(seq(0, max(simulation_results_poisso
                                                                    seq(500, max(simulation_results_pois

ggplot(simulation_results_poisson, aes(x = total_cost_bracket, y = variance)) +
  geom_boxplot() +
  labs(title = "Variance Across Total Cost Brackets",
       x = "Total Cost Bracket",
       y = "Variance") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        minimal = TRUE)
ggplot(simulation_results_poisson, aes(x = total_cost, y = mean_bias, color = as.factor(G), group = G))
  geom_line() +
  geom_point() +
  labs(title = "Mean Bias vs. Total Cost for Different G Values",
       x = "Total Cost",
       y = "Mean Bias",
       color = "Number of Clusters (G)") +
  theme_minimal()
```