Optimal Allocation of Clusters and Observations Under Budget Constraints in Experimental Designs for Cluster Randomised Trials

PHP2550 Project 3: A simulation study

Yanwei (Iris) Tong

2024-11-12 (Revised 2024-12-10)

Abstract

Purpose: Our study aims to determine the optimal number of clusters (G) and observations per cluster (R) under a fixed budget (B) to maximize the efficiency of treatment effect estimates in a cluster randomized trial. The budget constraint includes a higher cost for the initial observation in each cluster (c_1) compared to subsequent observations (c_2, where $c_2 < c_1$), simulating a realistic cost structure in study designs. We conduct simulations to explore different parameter settings, including cost ratios (c_1/c_2), within-cluster and between-cluster variances, and outcome distributions (Normal vs. Poisson). The goal is to find the configuration that yields the most precise estimate of the treatment effect while maintaining an efficient use of resources.

Methods: In this study, we conducted a series of simulation experiments under the ADEMP framework to investigate the impact of different design choices on the efficiency of treatment effect estimates under a fixed budget constraint (B). The number of clusters (G), was systematically iterated/varied to explore their effect on the variance of the estimated treatment effect ($\hat{\beta}$). For each value of G, the maximum feasible value of observations per cluster (R) was determined to remain within budget. We then simulated data 100 times for each combination of G and R and fitted hierarchical models and mixed-effect regressions to obtain estimates of $\hat{\beta}$. Additionally, we examined the influence of other parameters, such as within-cluster variance (σ^2) for Y~Normal, between-cluster variance (γ^2) for both Y~Normal and Poisson, and cost ratio structures c_1/c_2 to investigate their impacts on finding the optimal design. Key performance measures included variance, empirical bias, MSE, coverage probability, and power of the estimated treatment effects. This allowed us to derive insights into the trade-offs between cluster size and cluster composition for efficient resource allocation.

Results and conclusion: The simulation results showed that the optimal G and R are influenced by the relative costs (c_1/c_2) and the variance components (γ^2, σ^2). When the cost of adding clusters (c_1) is reduced or the cost of adding observations within clusters (c_2) is increased, the optimal design tends to favor more clusters with fewer observations per cluster. Reducing between-cluster variability (γ^2) generally leads to a preference for fewer clusters, while increasing within-cluster variability (σ^2) requires more clusters to minimize variance. Across different scenarios, minimizing the variance of the treatment effect estimate consistently led to high coverage probability and power, highlighting the robustness of the chosen optimal designs.

INTRODUCTION

Clustered experimental designs with cost constraints are commonly seen in real-world applications, such as energy consumption studies (e.g., Chen et al. (2018)), cloud computational resource allocation (e.g., Du et al. (2000)), and clinical trials/healthcare interventions (e.g., Breukelen and Candel (2012)). In these scenarios, the cost of recruiting the first subject in a cluster (e.g., setting

up a server or initiating an intervention) is often significantly higher than that of recruiting additional subjects. Understanding how to efficiently allocate resources under such budget constraints can help optimize decision-making processes and improve the statistical power and accuracy of the experiments. In this study, we aim to provide practical recommendations for researchers on how to allocate resources effectively in clustered settings to achieve precise and reliable treatment effect estimates.

In this project, we aim to explore the optimal design of cluster-randomized trials under budget constraints. Specifically, we investigate the effects of different numbers of clusters (G) and observations per cluster (R) on the precision of the estimated treatment effect (β), given a fixed budget (B) and different cost structures for sampling. Our approach uses a simulation-based methodology to determine the best combination of design parameters that minimizes the variance of the estimated treatment effect. We consider how varying key parameters, such as between-cluster and within-cluster variances, as well as the cost of sampling the first unit in each cluster (c_1) and subsequent units (c_2), affects the overall efficiency of the design.

ADEMP SIMULATION DESIGN

Aims and Objectives

- *Aim 1*: To evaluate the impact of different design choices (number of clusters G and observations per cluster R) under the budget constraint B and the fixed cluster cost for the first sample c_1 and the cost for all additional samples in the same cluster c_2 where $c_2 < c_1$.
- *Aim 2*: To explore the relationships between the underlying data generation mechanism parameters (e.g., variance components γ^2 and σ^2) and cost structure (relative costs c_1/c_2) and their impacts on estimation efficiency of treatment effects.
- *Aim 3*: To compare simulation performances and patterns under different outcome distributions (Normal vs. Poisson).

Data Generating Mechanisms

Budget constraint: The budget constraint is defined as $Gc_1 + G(R-1)c_2 \leq B$. This reflects that each cluster incurs a higher cost for the initial observation (c_1), while subsequent observations within the same cluster are less costly (c_2).

Hierarchical Model for Normal Outcomes For the Normal outcome model, the cluster-level outcomes are generated based on a linear model that incorporates both fixed treatment effects and random cluster effects, where the random effects introduce variability specific to each cluster. Subsequently, each observation within a cluster is generated with additional residual noise, capturing within-cluster variability. This hierarchical approach allows us to separately model both the variability between clusters and the variability within each cluster.

1) **Cluster-level mean:** The cluster-level mean (μ_i) depends on a fixed effect for treatment or control groups, as well as a random effect capturing the variability among clusters:

$$\mu_{i0} = \alpha + \beta X_i, \quad \mu_i \mid \epsilon_i = \mu_{i0} + \epsilon_i, \quad \epsilon_i \sim N(0, \gamma^2) \rightarrow \mu_i \sim N(\mu_{i0}, \gamma^2)$$

Here, X_i represents the treatment indicator ($0 = \text{control}, 1 = \text{treatment}$), and ϵ_i is the random effect associated with cluster i, representing unobserved cluster-level factors that affect the

outcome. γ^2 represents the variance of these cluster-level random effects.

2) **Observation-level outcomes:** Each observation (Y_{ij}) within cluster i is generated around the cluster mean (μ_i) with an additional random noise component:

$$Y_{ij} \mid \mu_i = \mu_i + e_{ij}, \quad e_{ij} \sim N(0, \sigma^2) \rightarrow Y_{ij} \mid \mu_i \sim N(\mu_i, \sigma^2)$$

The residual e_{ij} represents within-cluster variability, assumed to be normally distributed with variance σ^2.

3) **Marginal outcome distribution:** The marginal distribution of Y_{ij} given the treatment indicator X_i can be derived by integrating over the random effects. The marginal mean and variance are given by:

$$\text{Marginal mean:} \quad \mathbb{E}[Y_{ij} \mid X_i] = \alpha + \beta X_i, \text{ Marginal variance:} \quad Var(Y_{ij} \mid X_i) = \gamma^2 + \sigma^2$$

These expressions illustrate the overall variability in the data, which is composed of both between-cluster variance (γ^2) and within-cluster variance (σ^2).

Hierarchical Model for Poisson Outcomes For the Poisson outcome model, the cluster-level outcome is modeled on the logarithmic scale to ensure positivity, which is characteristic of count data. Each observation within a cluster is modeled as a Poisson random variable with a rate parameter determined by the cluster-level mean. The hierarchical structure is preserved by first modeling the cluster mean and then generating individual outcomes from that cluster-specific rate. This approach is useful in scenarios where outcomes are counts and are assumed to follow a Poisson distribution.

1) **Cluster-level mean:** The cluster-level rate parameter (μ_i) is modeled on the logarithmic scale to ensure non-negativity:

$$\log(\mu_i) = \alpha + \beta X_i + \epsilon_i \; \textit{where} \; \epsilon_i \sim N(0, \gamma^2), \; \mu_i \sim \text{LogNormal}(\alpha + \beta X_i, \gamma^2)$$

The exponentiated μ_i follows a log-normal distribution, which ensures that the rate parameter is always positive, as required for Poisson-distributed outcomes.

2) **Observation-level outcomes:** Each observation (Y_{ij}) within cluster i follows a Poisson distribution with the rate parameter μ_i:

$$Y_{ij} \mid \mu_i \sim \text{Poisson}(\mu_i)$$

This captures the individual-level variability, where the rate of occurrence is determined by the cluster-level mean.

3) **Summing within a cluster:** The aggregated outcome for cluster i is given by summing over all individual outcomes within the cluster:

$$Y_i = \sum_{j=1}^{R} Y_{ij} \quad \textit{and} \quad Y_i \mid \mu_i \sim \text{Poisson}(R\mu_i)$$

Since the sum of independent Poisson random variables remains Poisson, the total count within a cluster also follows a Poisson distribution, with rate parameter equal to R times the cluster-level mean.

Estimand

Average Treatment Effect (β) : The primary estimand of interest is to estimate the average treatment effect $\hat{\beta}$. This parameter captures the difference in outcomes between treatment and control groups. For the Normal outcome model, β represents the mean difference in outcomes, while in the Poisson outcome model, it represents the log-relative rate difference between treatment and control groups. And the ultimate objective related to the estimand is to optimize the study design to achieve precise estimates of *beta*. This involves **exploring different trade-offs between increasing G versus R**, given the differing costs. In other words, we would like to find the optimal combination of cluster size G and number of observations per cluster R that provides the most efficient allocation of resources while maintaining a budget constraint and thus minimizes the variance of $Var(\hat{\beta})$. More to be discussed in the "Performance Measures" subsection as follows.

Methods to Evaluate

The evaluation process involves conducting repeated simulations across a range of different design parameters, exploring how these parameters influence the variance of the estimated treatment effect ($\hat{\beta}$). Specifically, we aim to iterate through combinations of the number of clusters (G), and for each G, we calcualte the largest possible number of observations per cluster (R) under a fixed budget constraint (B). This involves:

1) **Iterating through all possible values of G**: - For each simulation scenario, we iterate G from $G = 2$ to $\lfloor B/c_1 \rfloor$, where c_1 is the cost of sampling the first unit in each cluster. For each value of G, we compute the corresponding R such that the cost remains within budget:

$$R = \left\lfloor \frac{B - G \cdot c_1}{G \cdot c_2} + 1 \right\rfloor$$

   where c_2 is the cost of additional samples within a cluster.

2) **Simulating 100 Times for Each Combination of G and R**: - For each combination of G and R, we conduct 100 repeated simulations to generate datasets and fit the mixed-effect regression models. This helps us estimate the variance of the treatment effect ($\text{Var}(\hat{\beta})$) for that specific design configuration.

3) **Optimization to find the optimal combination**: - After running through all possible combinations of G and R, the optimal combination is selected based on the variance of $\hat{\beta}$. Specifically, we aim to **find the combination of G and R that results in the smallest variance of $\hat{\beta}$**, indicating the most precise estimate under the given budget.

Also, as of *Aim2*, to understand the impact of various parameters on the optimal design:

1) **Vary c_1 and c_2**, especially considering different ratios of c_1/c_2, to determine how changes in cost structure influence the optimal allocation of resources.

2) **Vary σ^2 and γ^2**, representing within-cluster and between-cluster variance, respectively, to assess how different variance components affect the optimal design and the efficiency of the treatment effect estimation.

3) **Evaluate patterns and relationships**: Analyze how changes in cost structure and variance components contribute to changes in the optimal design. This helps us derive practical recommendations for designing cluster-randomized trials under varying budgetary and variance conditions.

Note: The budget is fixed at $B = 4000$ for all analyses, which would allow us to better investigate the impact of the cost structure (c_1/c_2) or the variance parameters (σ^2, γ^2) on the optimal experimental design. This approach ensures that any changes observed in the optimal design are due to the variation in these parameters rather than changes in the overall resource availability. Additionally, we have only considered two values for β (2 and 4), as our primary focus is on finding the optimal G and R combination that minimizes the variance of $\hat{\beta}$. Since the main goal is to achieve minimal variance, the specific value of β itself is less critical in this analysis.

Furthermore, the intercept parameter α is kept constant across all analyses. In the Normal case, α represents a baseline effect that does not influence the optimal allocation of G and R or the variance of $\hat{\beta}$. However, in the Poisson case, α impacts the overall variance due to the variance-mean relationship inherent to the Poisson distribution. Despite this, to better isolate the effects of parameters such as c_1/c_2, σ^2, γ^2, and β, and to avoid introducing additional complexity, α is held constant throughout the analyses. Since we are already varying β, keeping α fixed ensures consistency and comparability across scenarios.

Performance Measures

1) **Variance of Treatment Effect Estimates ($Var(\hat{\beta})$)**: The most important performance metrics for this study is the variance of the estimated treatment effect. The variance quantifies the precision of the estimates. In this study, the design parameters– such as the number of clusters (G) and the number of observations per cluster (R)– are varied to determine how they affect the variance of $\hat{\beta}$ under a fixed B, c_1, and c_2. The aim is to determine which strategy provides the most efficient use of the budget in terms of reducing the variance of the estimated treatment effect.

2) **Bias**: Bias measures the difference between the expected value of the estimated treatment effect ($\hat{\beta}$) and the true value (β). In this study design, essentially, $\hat{\beta}$ is an unbiased estimator, meaning that its expectation equals the true value: $\mathbb{E}(\hat{\beta}) = \beta$. However, in practice, issues such as singularity, rank deficiency, or convergence difficulties in model estimation might introduce practical bias. Therefore, although $\hat{\beta}$ is theoretically unbiased, we include bias as a performance metric to monitor and quantify any potential deviations that arise due to practical complexities during estimation.

3) **MSE**: Mean Squared Error (MSE) combines both variance and bias, providing a measure of the overall accuracy of $\hat{\beta}$. It is defined as: $MSE(\hat{\beta}) = Bias(\hat{\beta})^2 + Var(\hat{\beta})$. MSE captures both the precision and the accuracy of the treatment effect estimates. It helps identify designs that not only reduce the variance but also minimize the overall error in the estimates.

4) **Coverage Probability**: We evaluated the coverage probability of the estimated confidence intervals for the treatment effect. This metric assesses the proportion of confidence intervals that include the true value β. Mathematically, Coverage = $\Pr(\hat{\beta}_{\alpha\ \text{low}} \leq \beta_1 \leq \hat{\beta}_{\alpha\ \text{upp}})$, which is important for evaluating the reliability of the estimated treatment effect under different designs.

5) **Power**: Power is the probability of correctly detecting a treatment effect when one truly exists, and it depends on both the sample size and the magnitude of the effect. In this study, power is evaluated based on the proportion of times that the null hypothesis ($H_0 : \beta = 0$) is correctly rejected across the simulations. A higher power indicates greater sensitivity in detecting true effects, implying that the design is sufficiently powered to identify meaningful

differences between treatment groups.

RESULTS

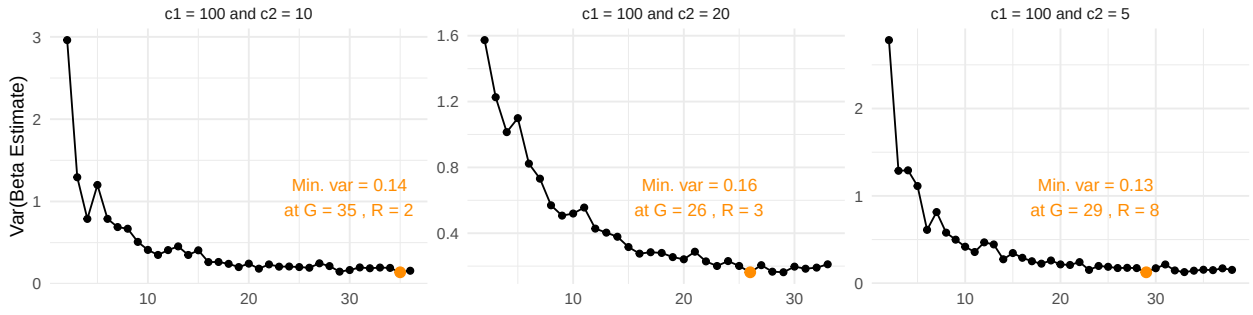
Investigating **Figure 1** horizontally presents the relationship between the variance of the beta estimate and G for different cost scenarios, with a fixed budget of $B = 4000$, $c1/c2 = 5, 10, 20$ and an outcome distribution following a normal model ($Y \sim$ Normal). The black line indicates the variance trend, while the orange point represents the configuration (combination of G and R) with the minimum variance. Each panel corresponds to a different combination of c_1 and c_2, showing the overall trend of variance reduction as the number of clusters (G) changes, and indicating the optimal design configuration. In all panels, we observe that the variance of the $\hat{\beta}$ initially decreases as G increases, reaches a minimum, and then either fluctuates or remains relatively stable as G continues to increase.

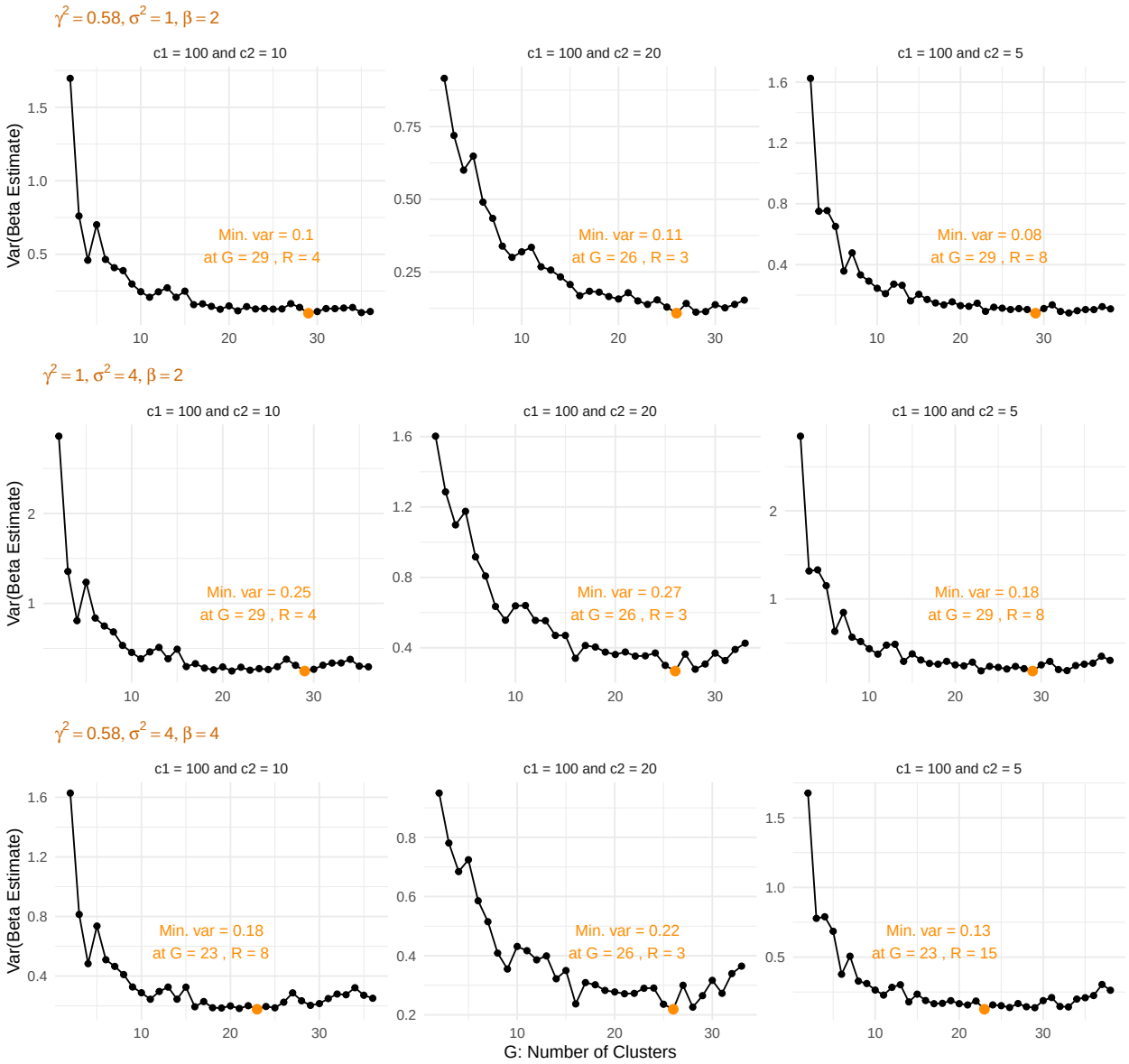
The optimal number of clusters that minimizes variance is influenced by the relative costs of adding clusters and adding observations within clusters. When the cost of adding observations within clusters (c_2) is high relative to the cost of adding clusters (c_1), the optimal configuration tends to favor more clusters (G) with fewer observations per cluster (R). Conversely, when the cost per observation (c_2) is reduced, adding more observations becomes less expensive, which may lead to a slightly lower number of clusters, though a higher number of clusters is still often preferred. Similarly, when the cost per cluster (c_1) is lower, the optimal design tends to favor more clusters, as the budget allows for a larger number of smaller clusters.

The observed pattern indicates that the optimal design for minimizing variance depends on the relative cost structure between c_1 and c_2. Higher c_1/c_2 ratios generally favor larger clusters (higher R), whereas higher c_2 tends to favor fewer clusters (G) with more observations. This highlights the trade-offs in resource allocation between expanding the number of clusters versus increasing the size of each cluster.

Figure 1 Relationship between Variance of the Beta Estimate and the Number of Clusters (B = 4000, Y ~ Normal)

$\gamma^2 = 1, \sigma^2 = 1, \beta = 4$





Examining the all four rows of **Figure 1** vertically reveals the relationship between different variance parameters (γ^2 and σ^2) and the optimal combination of G and R that minimize the variance of the treatment effect estimate ($\hat{\beta}$).

When γ^2 is higher, such as in the first and third rows, the optimal number of clusters tends to be higher. This is because greater between-cluster variability requires more clusters to capture this variability effectively. Lower σ^2, indicating less within-cluster variance, leads to a more precise estimate, resulting in an overall lower variance of $\hat{\beta}$ across the clusters.

Conversely, when σ^2 increases (seen in the third and fourth rows), the optimal number of clusters (G) becomes slightly lower compared to scenarios with lower σ^2. Increased within-cluster variability (σ^2) makes it more efficient to allocate the budget towards a balanced design between clusters (G) and observations per cluster (R), resulting in fewer clusters.

Figure 2 Relationship between Variance of the Beta Estimate and the Number of Clusters (B = 4000, Y ~ Poisson)

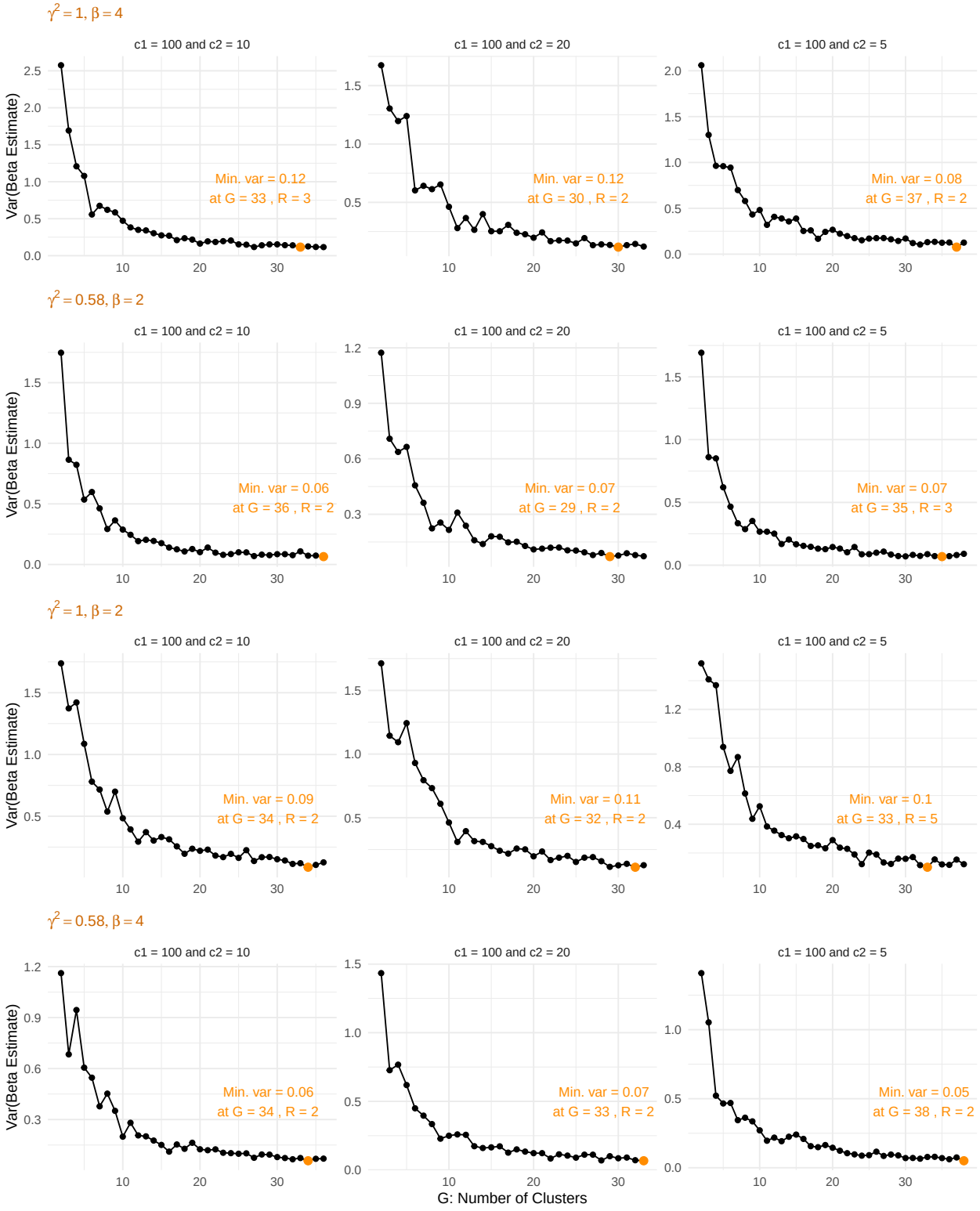


Figure 2 shows similar results for the Poisson models, with the optimal number of clusters (G) also being sensitive to changes in the cost structure (c_1 and c_2). Lower cost per observation (c_2) results in a slight reduction in the optimal number of clusters while favoring a higher number of

Table 1: Summary for Performance Measures of the Optimal Design under Different Data Generating Settings

Model	c1	c2	σ^2	γ^2	G	R	True β	Mean($\hat{\beta}$)	Var($\hat{\beta}$)	Bias	MSE	Coverage	Power
Normal	100	10	1	0.58	29	4	2	1.9612	0.0989	-0.0388	0.1004	0.96	1.00
	100	20	1	0.58	26	3	2	2.0055	0.1081	0.0055	0.1081	0.97	1.00
	100	5	1	0.58	29	8	2	1.9840	0.0808	-0.0160	0.0810	0.97	1.00
	100	10	4	1.00	29	4	2	1.9499	0.2482	-0.0501	0.2507	0.95	0.97
	100	20	4	1.00	26	3	2	2.0002	0.2679	0.0002	0.2679	0.98	0.92
	100	5	4	1.00	29	8	2	1.9790	0.1824	-0.0210	0.1829	0.96	0.99
	100	10	4	0.58	23	8	4	3.9778	0.1772	-0.0222	0.1777	0.97	1.00
	100	20	4	0.58	26	3	4	3.9953	0.2189	-0.0047	0.2189	0.99	1.00
	100	5	4	0.58	23	15	4	4.0584	0.1263	0.0584	0.1297	0.98	1.00
	100	10	1	1.00	35	2	4	3.9768	0.1394	-0.0232	0.1399	0.97	1.00
	100	20	1	1.00	26	3	4	4.0103	0.1626	0.0103	0.1627	0.97	1.00
	100	5	1	1.00	29	8	4	3.9790	0.1252	-0.0210	0.1257	0.98	1.00
Poisson	100	10	NA	0.58	36	2	2	1.9266	0.0649	-0.0734	0.0703	0.95	1.00
	100	20	NA	0.58	29	2	2	2.0080	0.0710	0.0080	0.0710	0.98	1.00
	100	5	NA	0.58	35	3	2	2.0021	0.0681	0.0021	0.0681	0.92	1.00
	100	10	NA	1.00	34	2	2	1.9998	0.0872	-0.0002	0.0872	0.95	1.00
	100	20	NA	1.00	32	2	2	2.0516	0.1110	0.0516	0.1137	0.98	1.00
	100	5	NA	1.00	33	5	2	1.9879	0.1007	-0.0121	0.1008	0.96	1.00
	100	10	NA	0.58	34	2	4	4.0258	0.0575	0.0258	0.0582	0.94	1.00
	100	20	NA	0.58	33	2	4	4.0486	0.0673	0.0486	0.0697	0.95	1.00
	100	5	NA	0.58	38	2	4	4.0261	0.0508	0.0261	0.0515	0.94	1.00
	100	10	NA	1.00	33	3	4	3.9766	0.1156	-0.0234	0.1161	0.91	1.00
	100	20	NA	1.00	30	2	4	3.9923	0.1166	-0.0077	0.1167	0.96	1.00
	100	5	NA	1.00	37	2	4	3.9295	0.0768	-0.0705	0.0818	0.97	1.00

observations per cluster (R). Conversely, reducing the cost per cluster (c_1) typically leads to an increase in the optimal G, as more clusters can be sampled within the fixed budget. When γ^2 is smaller, indicating reduced between-cluster variability, there is a greater tendency to allocate resources towards more clusters rather than adding observations within each cluster.

Overall, the results underscore the trade-off between capturing between-cluster variability effectively and optimizing resource allocation to minimize the variance of the treatment effect estimates. Compared to the Normal outcome models, the Poisson models exhibit lower sensitivity to cost changes and lower variability in the optimal number of clusters G.

Table 1 summarizes the performance measures of the optimal design across different data-generating settings for Normal and Poisson models. When the design is optimal—indicated by a relatively low variance of the estimated treatment effect ($Var(\hat{\beta})$)—the coverage probability and power are consistently high (both reaching or equal to 1 in all cases), which is an expected

outcome. A lower variance means that the estimates are more precise, reducing uncertainty, which in turn leads to narrower confidence intervals that are more likely to contain the true effect, hence achieving high coverage. Additionally, with greater precision, the study is more likely to detect a true effect when it exists, resulting in high power. This relationship between low variance, high coverage, and high power aligns well with theoretical expectations in experimental design and statistical inference. Moreover, the emparical bias is minimal across the designs, supporting the theoretical property that $\hat{\beta}$ is unbiased.

CONCLUSION

The results of our simulation study highlight the importance of balancing the number of clusters G and the number of observations per cluster R under varying cost structures to achieve optimal study designs. Across both Normal and Poisson outcome distributions, a consistent trend emerges: reducing the cost per cluster (c_1) or increasing the cost per observation (c_2) tends to favor designs with more clusters and fewer observations per cluster, thereby reducing variability in the estimates of the treatment effect. Additionally, increasing γ^2 (between-cluster variance) generally leads to a preference for more clusters to adequately capture this variability, while increasing σ^2 (within-cluster variance) tends to increase the variance of the estimated treatment effect without significantly altering the optimal cluster structure. These patterns suggest that careful adjustments to cost parameters and cluster-level variability can substantially improve design efficiency.

When the design is optimal—indicated by a relatively low variance of the estimated treatment effect—coverage probability and power are consistently high, reflecting strong statistical reliability. These results demonstrate that reducing the variance in the treatment effect estimate leads to increased robustness, as seen through higher coverage and power, while bias remains low or negligible. The analysis reveals that adjustments to c_1 and c_2 significantly influence the optimal design by favoring larger numbers of clusters, thereby minimizing variance. Overall, these findings underscore the importance of carefully considering cost structures and variance components in the design of cluster-randomized trials to achieve both efficiency and statistical robustness.

LIMITATIONS

One limitation of this study is the computational constraint that prevented us from exploring all possible combinations of c_1, c_2 and the variance parameters σ^2 and γ^2 in greater detail. The simulation process required substantial computational resources, making it unfeasible to comprehensively evaluate the impact of each parameter under all possible conditions. Also, the number of iterations executed was kept 100 in this study instead of a higher number due to the limited computational capacility of personal computers. Additionally, the limited scope of parameter settings may restrict the generalizability of the findings across different experimental designs. Future studies could benefit from employing advanced computational techniques or parallel processing and cloud computing to more thoroughly investigate the impact of varying these design parameters.

Data and Code Availability

This project is a collaboration with Dr. Zhijin Wu from the Department of Biostatistics at Brown University. Replication scripts and simulated data and results are available at https://github.com/YanweiTong-Iris/PHP2550-ProjectPortfolio/tree/main/Project%203.

Reference

Breukelen, G. J. van, and Candel, M. J. (2012), "Calculating sample sizes for cluster randomized trials: We can keep it simple and efficient!" *Journal of clinical epidemiology*, Elsevier, 65, 1212–1218.

Chen, Y., Xie, G., and Li, R. (2018), "Reducing energy consumption with cost budget using available budget preassignment in heterogeneous cloud computing systems," *IEEE Access*, IEEE, 6, 20572–20583.

Du, X., Zhang, X., and Zhu, Z. (2000), "Memory hierarchy considerations for cost-effective cluster computing," *IEEE Transactions on Computers*, IEEE, 49, 915–933.

Code Appendix

```
# to prevent scientific notation
options(scipen=999)

# Set up knit environment
knitr::opts_chunk$set(echo = FALSE,
                      message = FALSE,
                      warning = FALSE,
                      error = FALSE)

# Load necessary packages
library(tidyverse)
library(kableExtra)
library(knitr)
library(ggplot2)
library(gridExtra)
library(grid)
library(gtsummary)
library(gt)
library(patchwork)
library(knitcitations)
library(MASS)
library(RColorBrewer)
library(cowplot)
library(lme4)
library(doParallel)
library(foreach)


# Define data saving paths
data_path = paste0(here::here(), "/Project 3/simulated_data")
result_path = paste0(here::here(), "/Project 3/results")
#' @param alpha Intercept parameter (baseline mean).
#' @param beta Treatment effect.
#' @param gamma_sq Variance of cluster-level random effects.
#' @param sigma_sq Variance of observation-level random noise.
#' @param B Total budget available for the study (in dollars).
#' @param c1 Cost of the first sample in each cluster.
#' @param c2 Cost of additional samples in the same cluster (c2 < c1).
#' @param G Number of clusters
#' @param R Number of observations per cluster
#' @param n_sim Number of simulation iterations for tracking purposes
#' @return Data frame containing simulated data (Y, X) and the input parameters.
#' @export
simulate_clustered_data <- function(dist = "Normal",
                                    alpha,
```

```
                                    beta,
                                    gamma_sq,
                                    sigma_sq,
                                    B,
                                    c1,
                                    c2,
                                    G,
                                    R,
                                    n_sim = 1) {

X <- rbinom(G, size = 1, prob = 0.5)
epsilon <- rnorm(G, mean = 0, sd = sqrt(gamma_sq))

Y_list <- list()
cluster_id <- c()
observation_id <- c()
n_sim_id <- c()

if (dist == "Normal") {
  for (i in 1:G) {
    mu_i <- alpha + beta * X[i] + epsilon[i]
    e_ij <- rnorm(R, mean = 0, sd = sqrt(sigma_sq))
    Y_ij <- mu_i + e_ij

    Y_list[[i]] <- Y_ij
    cluster_id <- c(cluster_id, rep(i, R))
    observation_id <- c(observation_id, 1:R)
    n_sim_id <- c(n_sim_id, rep(n_sim, R))
  }
} else if (dist == "Poisson") {
  # Generate Y for Poisson distribution
  for (i in 1:G) {
    # Calculate mu_i based on log-normal model
    log_mu_i <- alpha + beta * X[i] + epsilon[i]
    mu_i <- exp(log_mu_i)

    # Generate R Poisson observations for cluster i
    Y_ij <- rpois(R, lambda = mu_i)

    Y_list[[i]] <- Y_ij
    cluster_id <- c(cluster_id, rep(i, R))
    observation_id <- c(observation_id, 1:R)
    n_sim_id <- c(n_sim_id, rep(n_sim, R))
  }
}

Y <- unlist(Y_list)
```

```r
  X_obs <- rep(X, each = R)
  data <- data.frame(Y, X_obs, cluster_id, observation_id, n_sim = n_sim_id)
  return(data)
}
#' Find optimal G and R to minimize variance
#'
#' @param alpha Intercept parameter (baseline mean).
#' @param beta Treatment effect.
#' @param gamma_sq Variance of cluster-level random effects.
#' @param sigma_sq Variance of observation-level random noise.
#' @param B Total budget available for the study (in dollars).
#' @param c1 Cost of the first sample in each cluster.
#' @param c2 Cost of additional samples in the same cluster (c2 < c1).
#' @param n_simulations Number of simulations to run for each G and R combination.
#' @print List with optimal G, optimal R, and minimum variance.
#' @return The result dataset with mean and var of Beta_hat for all G and R combination
#' @export
find_optimal_G_R <- function(dist = "Normal", alpha, beta, gamma_sq, sigma_sq,
                             B, c1, c2, n_simulations = 100) {
  min_variance <- Inf
  best_G <- 0
  best_R <- 0

  results <- data.frame(alpha = numeric(), beta = numeric(),
                        gamma_sq = numeric(), sigma_sq = numeric(),
                        B = numeric(), c1 = numeric(), c2 = numeric(),
                        G = integer(), R = integer(),
                        betahat_mean = numeric(), variance = numeric(),
                        bias = numeric(), coverage_prob = numeric(), power = numeric())

  for (G in 2:(floor((B / c1)) - 1)) {
    R = floor((B - G * c1) / (c2 * G)) + 1
    if (R > 1) {
      betahats <- numeric(n_simulations)
      t_z_values <- numeric(n_simulations)
      coverage <- numeric(n_simulations)

      data_list <- list()

      set.seed(46)
      for (sim in 1:n_simulations) {
        data <- simulate_clustered_data(dist = dist,
                                        alpha, beta,
                                        gamma_sq, sigma_sq,
                                        B, c1, c2,
                                        G, R, n_sim = sim)
        family = ifelse(dist == "Normal", "gaussian", "poisson")
```

```
    # Fit the model and handle convergence issues
    tryCatch({
      model <- glmer(Y ~ X_obs + (1 | cluster_id), family = family, data = data)

      # Get fixed ennn  ffect estimate
      betahats[sim] <- fixef(model)["X_obs"]

      # Calculate confidence interval and t-value if available
      ci <- tryCatch(confint(model, parm = "X_obs", level = 0.95), error = function(e) NA)

      if (!is.na(ci[1]) && !is.na(ci[2])) {
        # Check if true beta is within confidence interval (coverage probability)
        coverage[sim] <- ifelse(ci[1] <= beta & ci[2] >= beta, 1, 0)
      } else {
        coverage[sim] <- NA
      }

      # Extract z-value for power calculation
      t_z_values[sim] <- coef(summary(model))[2, 3]
    }, error = function(e) {
      # Handle model fitting failures by assigning NA
      betahats[sim] <- NA
      coverage[sim] <- NA
      t_z_values[sim] <- NA
    })
    data_list[[sim]] <- data
  }

  # Combine all simulated datasets for this G and R combination
  combined_data <- do.call(rbind, data_list)

  subfolder = paste0(data_path,
                     "/", dist, "_beta_", beta, "_gammasq_", gamma_sq,
                     ifelse(dist == "Normal", "_sigmasq_", ""), sigma_sq,
                     "_B_", B, "_c1_", c1, "_c2_", c2)

  if (!dir.exists(subfolder)){dir.create(subfolder)}

  saveRDS(combined_data, paste0(subfolder, "/simulated_data_G_", G, "_R_", R, ".rds"))

  # Remove NA values from betahats
  betahats <- betahats[!is.na(betahats)]
  valid_t_z_values <- t_z_values[!is.na(t_z_values)]
  valid_coverage <- coverage[!is.na(coverage)]

  if (length(betahats) > 0) {
    betahat_mean <- mean(betahats)
```

```
      betahat_variance <- var(betahats)

      # Calculate Bias
      bias <- betahat_mean - beta

      # Calculate Coverage Probability
      coverage_prob <- mean(valid_coverage, na.rm = TRUE)

      # Calculate Power (proportion of t-values greater than critical value, assuming t dist
      # Approximate t or z value for large n
      power <- mean(abs(valid_t_z_values) > 1.96, na.rm = TRUE)

      # Compare betahat_variance only if it is not NA
      if (!is.na(betahat_variance) && betahat_variance < min_variance) {
        min_variance <- betahat_variance
        best_G <- G
        best_R <- R
      }
    } else {
      betahat_mean <- NA
      betahat_variance <- NA
      bias <- NA
      coverage_prob <- NA
      power <- NA
    }

    if (dist == "Poisson"){
      sigma_sq <- NA
    }

    # Append results for this combination of G and R
    results <- rbind(
      results,
      data.frame(
        alpha = alpha,
        beta = beta,
        gamma_sq = gamma_sq,
        sigma_sq = sigma_sq,
        B = B,
        c1 = c1,
        c2 = c2,
        G = G,
        R = R,
        betahat_mean = betahat_mean,
        betahat_var = betahat_variance,
        bias = bias,
        coverage_prob = coverage_prob,
```

```
          power = power
        )
      )
    }
  }

  # Save the results
  result_save <- paste0(
    result_path,
    "/simulation_results_", dist, "_beta_", beta,
    "_gammasq_", gamma_sq,
    ifelse(dist == "Normal", paste0("_sigmasq_", sigma_sq), ""),
    "_B_", B, "_c1_", c1, "_c2_", c2, ".csv"
  )

  write.csv(
    results,
    result_save,
    row.names = FALSE
  )

  return(results)
}

#' Make the point plot for Var(Betahat) vs. G
#'
#' @param result The result dataset
#' @param group_name For faceted plotting purpose
#' @print The point plot with Var(Betahat) vs G.
var_vs_G_plot_by_facet <- function(results, group_name, dist,
                                   # for plotting purpose
                                   title.indicator = T,
                                   x.title.indicator = T) {

  # Filter results for the specified group
  group_results <- results %>% filter(group == group_name)

  # Find the point with the lowest betahat_var for each unique combination of c1 and c2
  lowest_points <- group_results %>%
    group_by(c1, c2) %>%
    filter(betahat_var == min(betahat_var)) %>%
    ungroup()

  beta = unique(group_results[["beta"]])
  B = unique(group_results[["B"]])
  c1 = unique(group_results[["c1"]])
  c2 = unique(group_results[["c2"]])
```

```
  gamma_sq = unique(group_results[["gamma_sq"]])
  sigma_sq = unique(group_results[["sigma_sq"]])

  if (dist == "Normal"){
    subtitle_text = bquote(list(gamma^2 == .(gamma_sq), sigma^2 == .(sigma_sq), beta ==.(beta)))
    fig_num = 1
  } else{
    subtitle_text = bquote(list(gamma^2 == .(gamma_sq), beta ==.(beta)))
    fig_num = 2
  }

  # Create the faceted plot
  plot = ggplot(group_results, aes(x = G, y = betahat_var)) +
    geom_line(color = "black") +
    geom_point(color = "black") +
    geom_point(data = lowest_points, aes(x = G, y = betahat_var), color = "darkorange", size =
    geom_text(data = lowest_points, aes(x = G, y = betahat_var,
                                        label = paste("Min. var =", round(betahat_var, 2),
                                                      "\n at G =", G, ", R =", R)),
              vjust = -1.5, hjust = 0.9, color = "darkorange", size = 3.5) +
    labs(
      title = paste("Figure ", fig_num, " Relationship between Variance of the Beta Estimate a
                    "Y ~ ", dist, ")"),
      subtitle = subtitle_text,
      x = "G: Number of Clusters",
      y = "Var(Beta Estimate)"
    ) +
      theme_minimal() +
    theme( plot.subtitle = element_text(size = 11, color = "darkorange3", face = "bold"))

  # Conditionally modify axis titles or plot title
  if (!x.title.indicator) {
    plot <- plot + theme(axis.title.x = element_blank())
  }

  if (!title.indicator) {
    plot <- plot + theme(plot.title = element_blank())
  }

  # Check if facet_label column exists and has at least one unique value before applying facet_
  if ("facet_label" %in% colnames(group_results) && n_distinct(group_results$facet_label) > 0)
    plot <- plot +
      facet_wrap(~ facet_label, scales = "free")
    print(plot)
  }

}
```

```r
alpha_list <- rep(4, 12)
beta_list <- c(4, 4, 4, 2, 2, 2, 2, 2, 2, 4, 4, 4)
gamma_sq_list <- c(1, 1, 1, 0.58, 0.58, 0.58, 1, 1, 1, 0.58, 0.58, 0.58)
sigma_sq_list <- c(1, 1, 1, 1, 1, 1, 4, 4, 4, 4, 4, 4)
B_list <- rep(4000, 12)
c1_list <- c(100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100)
c2_list <- c(5, 10, 20, 5, 10, 20, 5, 10, 20, 5, 10, 20)

num_cores <- detectCores() - 2
cl <- makeCluster(num_cores)
registerDoParallel(cl)

Normal_combined_results <- foreach(i = 1:12, .combine = rbind, .packages = c("lme4", "dplyr")) %dopar% {
  alpha <- alpha_list[i]
  beta <- beta_list[i]
  gamma_sq <- gamma_sq_list[i]
  sigma_sq <- sigma_sq_list[i]
  B <- B_list[i]
  c1 <- c1_list[i]
  c2 <- c2_list[i]

  # Call find_optimal_G_R function
  result <- find_optimal_G_R(dist = "Normal", alpha = alpha, beta = beta,
                             gamma_sq = gamma_sq, sigma_sq = sigma_sq,
                             B = B, c1 = c1, c2 = c2,
                             n_simulations = 100)

  # Add the group to the result for plotting
  result$group <- paste0("Group ", ceiling(i / 3))

  # Combine c1 and c2 to create a single label for facetting
  result$facet_label <- paste0("c1 = ", c1, " and c2 = ", c2)

  return(result)
}

stopCluster(cl)
var_vs_G_plot_by_facet(Normal_combined_results,
                       group_name = "Group 1",
                       dist = "Normal",
                       title.indicator = T,
                       x.title.indicator = F)
var_vs_G_plot_by_facet(Normal_combined_results,
                       group_name = "Group 2",
                       dist = "Normal",
                       title.indicator = F,
                       x.title.indicator = F)
```

```r
var_vs_G_plot_by_facet(Normal_combined_results,
                       group_name = "Group 3",
                       dist = "Normal",
                       title.indicator = F,
                       x.title.indicator = F)
var_vs_G_plot_by_facet(Normal_combined_results,
                       group_name = "Group 4",
                       dist = "Normal",
                       title.indicator = F,
                       x.title.indicator = T)
num_cores <- detectCores() - 2
cl <- makeCluster(num_cores)
registerDoParallel(cl)

Poisson_combined_results <- foreach(i = 1:12, .combine = rbind, .packages = c("lme4", "dplyr"))
  alpha = alpha_list[i]
  beta = beta_list[[i]]
  gamma_sq = gamma_sq_list[i]
  sigma_sq = sigma_sq_list[i]
  B = B_list[i]
  c1 = c1_list[i]
  c2 = c2_list[i]
  result <- find_optimal_G_R(dist = "Poisson", alpha = alpha, beta = beta,
                             gamma_sq = gamma_sq, sigma_sq = sigma_sq,
                             B = B, c1 = c1, c2 = c2,
                             n_simulations = 100)

  # Add the group to the result for plotting
  result$group <- paste0("Group ", ceiling(i / 3))

  # Combine c1 and c2 to create a single label for facetting
  result$facet_label <- paste0("c1 = ", c1, " and c2 = ", c2)

  return(result)
}
stopCluster(cl)


var_vs_G_plot_by_facet(Poisson_combined_results,
                       group_name = "Group 1",
                       dist = "Poisson",
                       title.indicator = T,
                       x.title.indicator = F)
var_vs_G_plot_by_facet(Poisson_combined_results,
                       group_name = "Group 2",
                       dist = "Poisson",
                       title.indicator = F,
```

```
                        x.title.indicator = F)
var_vs_G_plot_by_facet(Poisson_combined_results,
                        group_name = "Group 3",
                        dist = "Poisson",
                        title.indicator = F,
                        x.title.indicator = F)
var_vs_G_plot_by_facet(Poisson_combined_results,
                        group_name = "Group 4",
                        dist = "Poisson",
                        title.indicator = F,
                        x.title.indicator = T)
# Function to select the row with the minimum betahat_var for each group
select_min_var_rows <- function(results) {
  results %>%
    group_by(group, facet_label) %>%
    filter(!is.na(betahat_var)) %>%
    slice_min(betahat_var, with_ties = FALSE) %>%  # Select the row with the minimum betahat_v
    ungroup()
}

# Selecting rows with the minimum betahat_var for each group
normal_min_var <- select_min_var_rows(Normal_combined_results)
poisson_min_var <- select_min_var_rows(Poisson_combined_results)

# Adding MSE = bias^2 + var(beta)
normal_min_var <- normal_min_var %>%
  mutate(MSE = bias^2 + betahat_var)

poisson_min_var <- poisson_min_var %>%
  mutate(sigma_sq = NA,  # Set sigma_sq to NA for Poisson results
         MSE = bias^2 + betahat_var)

# Select relevant columns
normal_selected <- normal_min_var %>%
  mutate(model = "Normal") %>%
  dplyr::select(model, c1, c2, sigma_sq, gamma_sq, G, R, beta, betahat_mean,
                betahat_var, bias, MSE, coverage_prob, power)

poisson_selected <- poisson_min_var %>%
  mutate(model = "Poisson") %>%
  dplyr::select(model, c1, c2, sigma_sq, gamma_sq, G, R, beta, betahat_mean,
                betahat_var, bias, MSE, coverage_prob, power)

# Combine Normal and Poisson results into one table
combined_results <- bind_rows(
  normal_selected,
  poisson_selected
```

```
)

# Arrange the results so that Normal rows come first, followed by Poisson rows
combined_results <- combined_results %>%
  arrange(model, beta, gamma_sq)

# Format numeric values to 4 decimal places
combined_results <- combined_results %>%
  mutate(across(where(is.numeric), ~ round(.x, 4)))

# Create a kable table with row groupings for "Y ~ Normal" and "Y ~ Poisson"
kable_table <- combined_results %>%
  knitr::kable("latex",
    caption = "Summary for Performance Measures of the Optimal Design under Different Data Gene
    col.names = c(
      "Model",
      "c1",
      "c2",
      "$\\sigma^2$",
      "$\\gamma^2$",
      "G",
      "R",
      "True $\\beta$",
      "$\\mathrm{Mean}(\\hat{\\beta})$",
      "$\\mathrm{Var}(\\hat{\\beta})$",
      "Bias",
      "MSE",
      "Coverage",
      "Power"
    ), escape = F,
    booktabs = T) %>%
  kable_styling(full_width = FALSE,
                bootstrap_options = c("striped", "hover"),
                font_size = 9) %>%
  column_spec(1, bold = TRUE) %>%
  collapse_rows(columns = 1, valign = "top")

# Print the table (in R Markdown, this will render the table)
kable_table
```