

Designing Optimal Cluster-Randomized Trials

Dingxuan Zhang

December 2024

Abstract

Cluster randomized trials are a common design for evaluating treatment effects when observations are naturally grouped into clusters. In this study, we aim to determine the optimal allocation of resources in such trials under budget constraints. Specifically, we explore how the interplay between underlying data generation parameters (α , β , γ^2 , σ^2) and relative costs (c_1/c_2) influences the optimal choice of cluster number (G) and within-cluster sample size (R). Using a hierarchical model where outcomes are normally distributed, we develop a simulation study within the ADEMP framework that integrates estimation, performance metrics, and cost considerations.

Our results demonstrate that inter-cluster variability (γ^2) and intra-cluster variability (σ^2) have distinct impacts on the optimal design, with high γ^2 favoring increased G and high σ^2 prioritizing larger R . Additionally, the interaction effects of α (baseline mean) and β (treatment effect) significantly alter resource allocation, particularly under varying cost ratios. By systematically evaluating these factors, we establish clear guidelines for designing efficient cluster randomized trials under budget constraints.

Lastly, we extend our study to address Poisson-distributed outcomes, highlighting the generalizability of our approach and the potential differences in design strategies under alternative outcome distributions. Our findings provide practical insights into balancing statistical precision with resource limitations in real-world trial design.

Introduction

Cluster randomized trials are widely used in settings where observations are naturally grouped into clusters, such as schools, hospitals, or geographic regions. In such trials, clusters are randomly assigned to treatment or control groups, and all observations within a cluster receive the same treatment assignment. While this design simplifies randomization logistics, it introduces statistical challenges, particularly due to intra-cluster correlations. This makes the allocation of resources—such as the number of clusters (G) and the number of observations within each cluster (R)—critical for efficient study design.

A key constraint in real-world studies is the limited availability of resources, often represented by a fixed budget (B). Sampling a new cluster incurs a cost (c_1), and adding observations within an existing cluster costs less ($c_2 < c_1$). The relative cost ratio (c_1/c_2) greatly impacts the optimal choice of G and R . Additionally, the variance between clusters (γ^2) and within clusters (σ^2) determines the precision of treatment effect estimates, further complicating the design process.

In our simulation study, we shall address the following aims:

AIM 1: Design a simulation study using the ADEMP framework from class to evaluate potential study designs.

AIM 2: Explore relationships between the underlying data generation mechanism parameters and the relative costs c_1/c_2 and how these impact the optimal study design.

AIM3: Extend your simulation study to the setting in which Y follows a Poisson distribution with mean μ and explore how this impacts the results. The hierarchical model for this setting is given below.

Our study addresses these challenges by combining first two aims within a ADEMP framework then extend to aim 3 following these two steps:

1. Evaluate the impact of key parameters $(\alpha, \beta, \gamma^2, \sigma^2)$ and the cost ratio (c_1/c_2) on the optimal allocation of G and R .
2. Develop a simulation to analyze these factors under budget constraints.

ADEMP framework for normally distributed case

As we mentioned, we will combine first two aims in a single ADEMP framework, which is under the setting where our Y is normally distributed.

Aims

The primary aim of this study is to determine the optimal allocation of resources in a cluster randomized trial under budget constraints. Specifically, we focus on the following objectives:

1. Simulate to check how varying data generation parameters $(\alpha, \beta, \gamma^2, \sigma^2)$ will change the pattern of varying number of clusters (G) and the number of observations per cluster (R) when no budget constraint is set.
2. Evaluate the impact of the underlying data generation parameters on the optimal choice of the number of clusters (G) and the number of observations per cluster (R) when considering budget limit.
3. Investigate how the relative cost ratio (c_1/c_2) influences the trade-off between increasing G and R , given a fixed budget (B).

By addressing these aims, we aim to provide practical recommendations for balancing statistical precision with resource limitations in cluster randomized trials. This framework will focus on normally distributed outcomes, with an extension to Poisson-distributed outcomes discussed later in the study.

Data-Generating Mechanism

We use a hierarchical model to simulate data in the context of a cluster randomized trial. The primary goal is to evaluate how the underlying data generation parameters $(\alpha, \beta, \gamma^2, \sigma^2)$ and the relative cost ratio (c_1/c_2) influence the optimal design choices for the number of clusters (G) and the number of observations per cluster (R) under a fixed budget (B).

Hierarchical Model

1. **Cluster-level treatment assignment:** Each cluster $i = 1, \dots, G$ is randomly assigned to either the treatment group ($X_i = 1$) or the control group ($X_i = 0$).
2. **Cluster mean (μ_i):**

$$\mu_i \sim N(\alpha + \beta X_i, \gamma^2)$$

- α : Baseline mean for the control group.
- β : Treatment effect (difference in means between treatment and control groups).
- γ^2 : Between-cluster variance, representing heterogeneity among clusters.

3. **Individual-level observations** (Y_{ij}): Within each cluster i , $j = 1, \dots, R$ observations are generated as:

$$Y_{ij} \mid \mu_i \sim N(\mu_i, \sigma^2)$$

- σ^2 : Within-cluster variance, representing individual-level variability or measurement error.

Budget and Cost Constraints

The total budget B constrains the allocation of resources:

$$B = G \cdot c_1 + G \cdot (R - 1) \cdot c_2$$

- c_1 : Cost of sampling a new cluster. - c_2 : Cost of adding an additional observation within an existing cluster ($c_2 < c_1$).

Parameters for Simulation

- **Budget parameters:**
 - Budget: B .
 - Cost ratio: c_1/c_2 .
- **Varying parameters:**
 - G : Number of clusters.
 - R : Number of observations per cluster.
 - α : Baseline mean.
 - β : Treatment effect.
 - γ^2 : Between-cluster variance.
 - σ^2 : Within-cluster variance.

This hierarchical structure provides the foundation for evaluating the impact of these parameters on design choices and statistical efficiency. In our study, we will always assign the clusters to treatment group and control group with equal chance. The following is the head of a sample simulation where we set $G=10$, $R=5$, $\alpha = 1$, $\beta = 0.5$, $\gamma^2 = 0.5$, and $\sigma^2 = 1.0$.

##	cluster	treatment	observation	Y
## 1	1	0	1	3.9996472
## 2	1	0	2	2.7105846
## 3	1	0	3	0.2461169
## 4	1	0	4	2.9140900
## 5	1	0	5	1.7399427
## 6	2	1	1	0.7580933

Estimand

The primary parameter of interest in this study is the treatment effect (β) in a hierarchical data structure. To estimate β , we used a linear mixed-effects model, where treatment was modeled as a fixed effect and clusters as random effects. The model is specified as:

$$Y_{ij} = \alpha + \beta \cdot X_i + u_i + e_{ij}, \quad u_i \sim N(0, \gamma^2), \quad e_{ij} \sim N(0, \sigma^2),$$

where: - Y_{ij} is the outcome for observation j in cluster i , - X_i is the treatment assignment for cluster i ($0 = \text{control}, 1 = \text{treatment}$), - α is the intercept, - β is the fixed effect of interest, - u_i is the random effect for cluster i , - e_{ij} is the residual error.

The treatment effect (β) was estimated using the `lmer` function from the `lme4` package in R, which fits linear mixed-effects models via maximum likelihood. For each simulation scenario, the model was applied to simulated datasets, and $\hat{\beta}$ was extracted as the estimate of the treatment effect.

To evaluate the precision and variability of $\hat{\beta}$ across simulations, we focused on several performance metrics which will be introduced in the next section.

The hierarchical model was chosen for its ability to account for both between-cluster variability (γ^2) and within-cluster variability (σ^2), which are key factors influencing the precision of $\hat{\beta}$. By incorporating random effects, the model appropriately partitions the total variability in Y_{ij} into these components, ensuring an unbiased and efficient estimation of the fixed effect (β).

Method

In our study, we will evaluate our model performance based on several metrics. Then, focusing on these metrics, we will achieve our listed aims by comparing how different settings changed the performance.

Performance Metrics To evaluate the quality of the treatment effect estimates (β), we computed the following metrics across simulated datasets:

1. **Bias:** Bias measures the systematic difference between the average estimated treatment effect ($\bar{\hat{\beta}}$) and the true value (β). A smaller bias indicates that the estimator is centered closer to the true value.

$$\text{Bias} = \bar{\hat{\beta}} - \beta$$

2. **Variance:** Variance reflects the spread of the estimated treatment effects ($\hat{\beta}$) across simulations. High variance suggests that the estimator is unstable.

$$\text{Var}(\hat{\beta}) = \frac{\sum_{i=1}^n (\hat{\beta}_i - \bar{\hat{\beta}})^2}{n - 1}$$

3. **Mean Squared Error (MSE):** MSE is a comprehensive metric that combines Bias and Variance to quantify the overall error of the estimator.

$$\text{MSE} = \text{Bias}^2 + \text{Var}(\hat{\beta})$$

4. **Empirical Standard Error (Empirical SE):** Empirical SE represents the actual sampling variability of the estimates ($\hat{\beta}$) across simulations. It is calculated as the standard deviation of the estimates:

$$\text{Empirical SE} = \sqrt{\text{Var}(\hat{\beta})}.$$

Empirical SE serves as the primary metric for evaluating precision because it directly reflects the observed variability of the estimates under different configurations.

5. **Average Standard Error (Avg SE):** Avg SE is the mean of the standard errors reported by the model during each simulation. This metric provides a theoretical estimate of the variability based on model assumptions.

These metrics collectively provide a detailed assessment of how different configurations of clusters (G) and observations per cluster (R) influence the estimation performance under varying data generation scenarios and budget constraints. Below we generate the value of these metrics under one setting.

G	R	gamma2	sigma2	Bias	Variance	MSE	Empirical_SE	Avg_SE
10	5	0.5	1	0.0652877	0.2848406	0.2891031	0.5337046	0.543091

Analysis Steps

To conduct our analysis, we will follow a few steps. To start with, we would like to get an insight into the trend of how varying values of underlying data generation mechanism parameters will change the impact of G and R when no budget constraint is set. Then, we will conduct a grid search to see what is the optimal values of G and R under certain setting and we shall explain the result by our previously observed trends. Finally, we will moderate the ratio of costs and the budget to see how it will affect the result of grid search and present some visualizations.

Performance

Trend of varing G and R under different variance settings

Under this normally distributed case, the underlying parameters of our interest should be γ^2 and σ^2 since γ and σ is not affecting the variance of the result under normal distribution case (which is different from in the Poisson case). Therefore, to analyze the trend, we will consider in three different cases, where gamma is significantly greater than sigma, sigma is greater than gamma, and when they are all equal to 1. For these three settings, we create one table for varying G and a table for varying R.

Table 2: Performance Evaluation with fixed G

G	R	gamma2	sigma2	Bias	Variance	MSE	Empirical_SE	Avg_SE
20	5	10.0	0.1	-0.3824684	2.2221166	2.3683987	1.4906766	1.4695049
20	20	10.0	0.1	0.1178937	2.1640768	2.1779757	1.4710801	1.4646561
20	50	10.0	0.1	0.1727026	2.5098967	2.5397229	1.5842654	1.4794382
20	5	0.1	10.0	0.0099439	0.3930406	0.3931395	0.6269295	0.6933383
20	20	0.1	10.0	-0.0380752	0.1193874	0.1208372	0.3455249	0.3663980
20	50	0.1	10.0	-0.0313948	0.0795698	0.0805554	0.2820812	0.2499818
20	5	1.0	1.0	0.0077776	0.2509910	0.2510515	0.5009900	0.4881683
20	20	1.0	1.0	-0.0066360	0.1967219	0.1967660	0.4435334	0.4716393
20	50	1.0	1.0	-0.0304041	0.2050142	0.2059386	0.4527849	0.4652312

Table 3: Performance Evaluation with fixed R

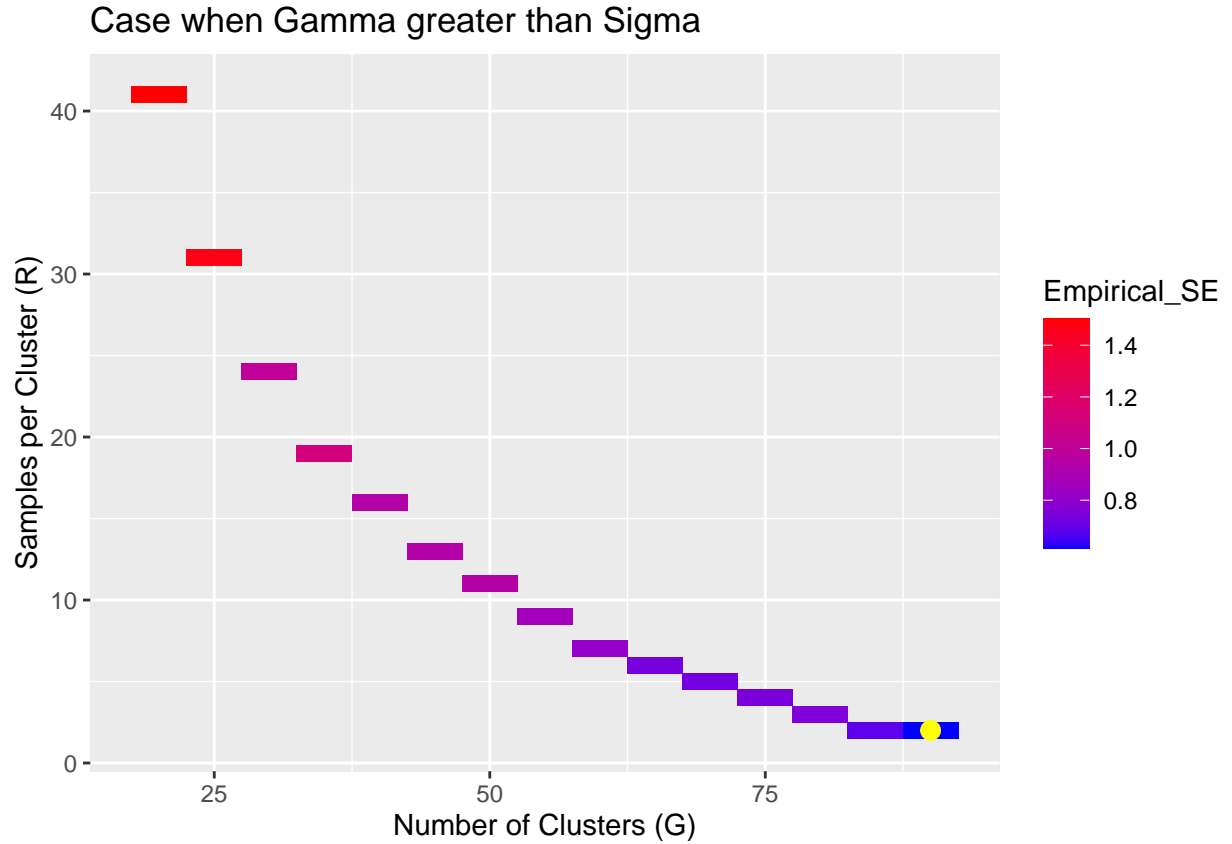
G	R	gamma2	sigma2	Bias	Variance	MSE	Empirical_SE	Avg_SE
20	20	10.0	0.1	-0.0411366	2.2742893	2.2759816	1.5080747	1.3834067
30	20	10.0	0.1	0.0204109	1.5266536	1.5270702	1.2355783	1.1787299
50	20	10.0	0.1	0.0743268	0.7500123	0.7555367	0.8660325	0.9026149
20	20	0.1	10.0	0.0214809	0.1574936	0.1579551	0.3968547	0.3583804
30	20	0.1	10.0	0.0318909	0.0744194	0.0754364	0.2727992	0.2877973
50	20	0.1	10.0	-0.0187751	0.0379653	0.0383178	0.1948468	0.2194562
20	20	1.0	1.0	-0.0225560	0.2432830	0.2437918	0.4932373	0.4643195
30	20	1.0	1.0	0.0065258	0.1558521	0.1558947	0.3947811	0.3795524
50	20	1.0	1.0	0.0188700	0.0823700	0.0827261	0.2870018	0.2898782

The results reveal patterns as expected. Throughout the project, we will mainly be evaluating performance based on variance and empirical SE. When the between-cluster variance (γ^2) is large and the within-cluster

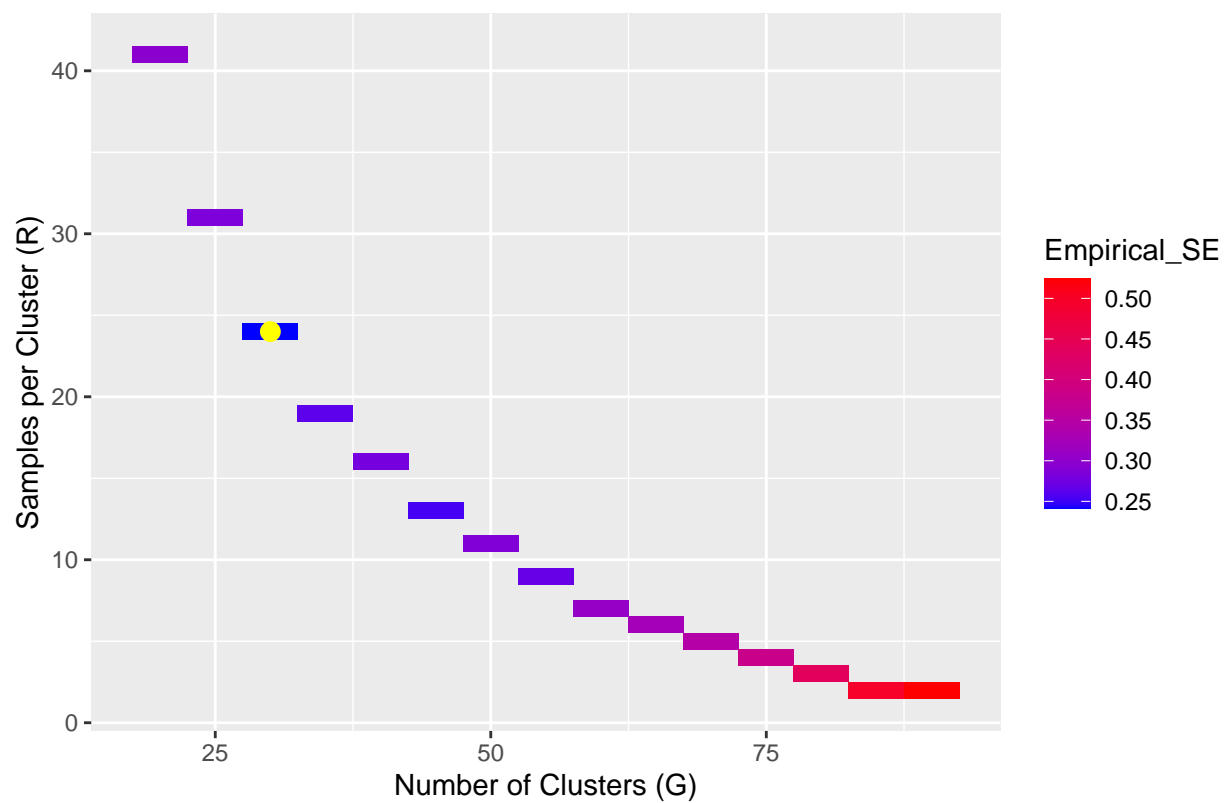
variance (σ^2) is small, increasing G significantly reduces the Variance and empirical SE, highlighting the importance of adding more independent clusters. Conversely, when σ^2 dominates, increasing R substantially improves precision by reducing the noise within clusters. Under balanced conditions ($\gamma^2 = \sigma^2 = 1$), both G and R contribute comparably to reducing variability, suggesting a need for balanced resource allocation. Empirical SE, as a direct measure of observed variability, aligns with these trends and serves as a key metric for evaluating the design's precision. These findings show the importance of determining the allocation of G and R based on the underlying data characteristics to achieve optimal experimental precision. Then, we will move on to the setting where budget is applied.

Budget constraint analysis using grid search

Now we will consider the case where budget is limited. We shall start with a setting where $B=1000$, $c_1=10$, and $c_2=1$. The grid search will start from $G=20$ with a step size of 5. As we know the trend from the last part, we will maximize R when G is fixed.



Case when Gamma less than Sigma



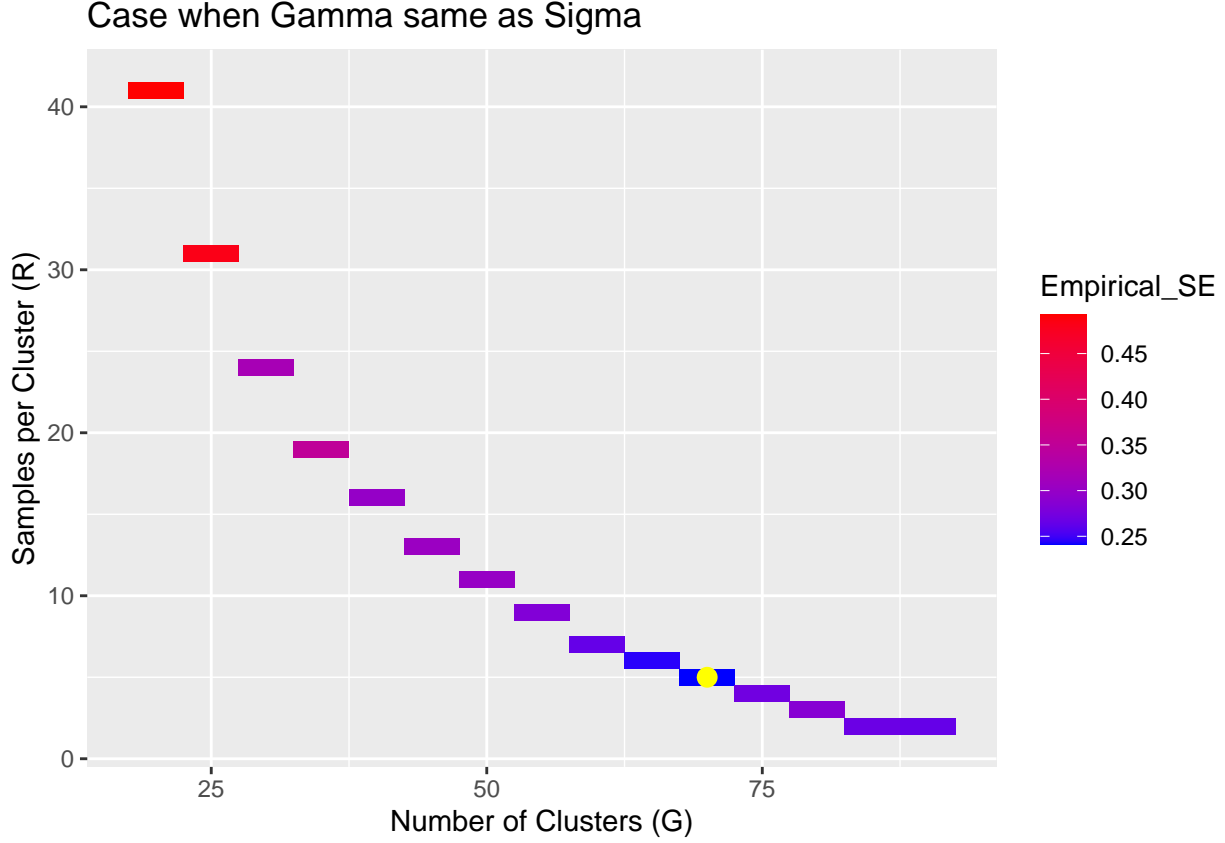


Table 4: Grid Search Results for Budget Allocation

G	R	gamma2	sigma2	Bias	Variance	MSE	Empirical_SE	Avg_SE
20	41	1	1	-0.0721777	0.2427380	0.2479476	0.4926845	0.4639972
25	31	1	1	0.0066852	0.2325320	0.2325767	0.4822157	0.4068861
30	24	1	1	-0.0486551	0.1013539	0.1037212	0.3183613	0.3756586
35	19	1	1	0.0516227	0.1222834	0.1249483	0.3496904	0.3484613
40	16	1	1	-0.0127214	0.0895321	0.0896939	0.2992191	0.3323643
45	13	1	1	0.0581820	0.0927732	0.0961583	0.3045869	0.3039914
50	11	1	1	0.0156089	0.0907943	0.0910379	0.3013209	0.3002701
55	9	1	1	0.0176984	0.0801825	0.0804957	0.2831652	0.2863881
60	7	1	1	-0.0351391	0.0701177	0.0713524	0.2647974	0.2793488
65	6	1	1	0.0271542	0.0601069	0.0608442	0.2451671	0.2730366
70	5	1	1	-0.0185136	0.0582490	0.0585917	0.2413482	0.2628112
75	4	1	1	-0.0215675	0.0739175	0.0743827	0.2718778	0.2613798
80	3	1	1	0.0231083	0.0824489	0.0829829	0.2871391	0.2613254
85	2	1	1	-0.0232507	0.0717779	0.0723185	0.2679141	0.2655348
90	2	1	1	0.0004980	0.0702180	0.0702183	0.2649869	0.2589228

The grid search results provide further insights into the optimal allocation of clusters (G) and observations per cluster (R) across different variance scenarios. In the first scenario, where between-cluster variance ($\gamma^2 = 10$) dominates and within-cluster variance ($\sigma^2 = 0.1$) is small, increasing G significantly reduces Variance and Empirical SE. This pattern demonstrates that allocating more budget to increasing the number of clusters effectively captures the diversity between groups and reduces overall variability. The optimal configuration in this scenario is $G = 90$ and $R = 2$, achieving the lowest Empirical SE of 0.6131. This finding underscores

the importance of prioritizing the number of clusters (G) when between-cluster differences are the primary source of variability.

In the second scenario, where within-cluster variance ($\sigma^2 = 10$) dominates and between-cluster variance ($\gamma^2 = 0.1$) is small, increasing R proves to be more effective in reducing noise and improving precision. The results show that $G = 20$ and $R = 41$ yield the lowest Empirical SE of 0.2973. This indicates that, in high within-cluster variability scenarios, increasing the number of observations per cluster (R) allows the model to average out the noise within each cluster, leading to better precision in estimating the treatment effect.

In the balanced variance scenario ($\gamma^2 = \sigma^2 = 1$), both G and R contribute comparably to reducing variability. Neither parameter alone is sufficient to achieve optimal precision, and a balanced allocation of resources is required. The optimal configuration is $G = 70$ and $R = 5$, resulting in the lowest Empirical SE of 0.2413. This setting highlights the need for a trade-off between the number of clusters and the number of observations per cluster when the between-cluster and within-cluster variances are balanced.

Overall, these results confirm that the dominant source of variability (γ^2 or σ^2) indicates the priority in budget allocation. When γ^2 dominates, increasing G should be prioritized, whereas when σ^2 dominates, increasing R is more effective. In cases where variances are balanced, a hybrid strategy that allocates resources to both G and R achieves the best results. Empirical SE serves as a reliable metric to guide these decisions, consistently identifying the configurations that yield the highest precision.

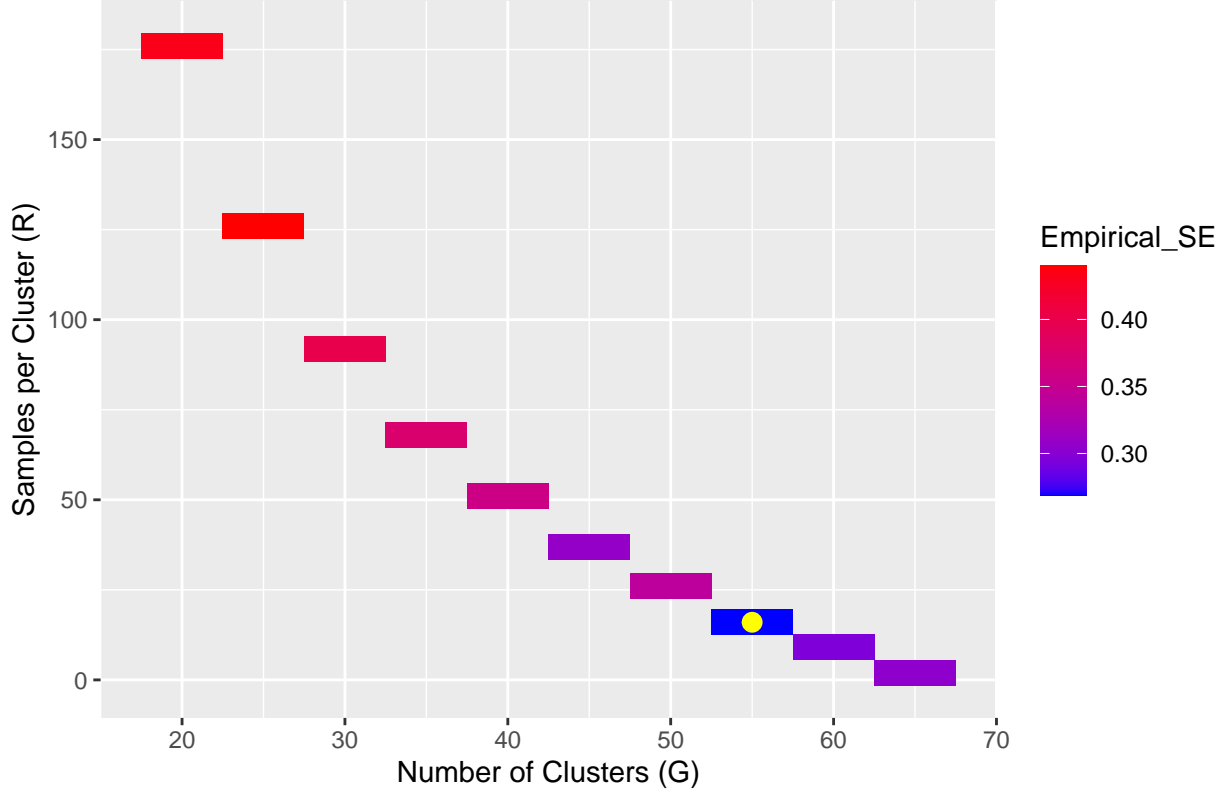
Change in cost ratio

Now that we alter the ratio of $c1/c2$ and check the result. In the following case, we increase $c1$ to 15 while lower $c2$ to 0.2 which means the ratio of $c1/c2$ is increased. Notice in the table the original optimal ratio of G/R was $70/5=14$, while the updated ratio is $55/16$. We can thus conclude that increase in $c1/c2$ ratio results in decrease of optimal G/R ratio.

Table 5: Grid Search Results for Budget Allocation

G	R	gamma2	sigma2	Bias	Variance	MSE	Empirical_SE	Avg_SE
20	176	1	1	-0.0001937	0.1868408	0.1868408	0.4322508	0.4537364
25	126	1	1	-0.0280598	0.1937857	0.1945731	0.4402110	0.4030164
30	92	1	1	-0.0180615	0.1599585	0.1602847	0.3999481	0.3760678
35	68	1	1	-0.0398807	0.1400793	0.1416697	0.3742717	0.3472913
40	51	1	1	0.0634428	0.1277816	0.1318066	0.3574655	0.3163845
45	37	1	1	0.0594099	0.0945157	0.0980452	0.3074340	0.3042317
50	26	1	1	-0.0219699	0.1162556	0.1167382	0.3409627	0.2842778
55	16	1	1	-0.0230111	0.0722939	0.0728234	0.2688752	0.2788945
60	9	1	1	0.0080801	0.0874851	0.0875504	0.2957788	0.2759136
65	2	1	1	0.0191751	0.0931277	0.0934954	0.3051684	0.3038443

Case when Gamma same as Sigma



Extension

We shall now extend our normally distributed model to the Poisson model.

Data Generation for Poisson Model

To simulate data under the hierarchical Poisson model, we assumed the following structure:

$$\log(\mu_i) \sim N(\alpha + \beta X_i, \gamma^2), \quad Y_{ij} \sim \text{Poisson}(\mu_i),$$

where: - X_i is the treatment assignment for cluster i (0 = control, 1 = treatment), - μ_i is the mean outcome for cluster i , - α is the intercept, - β is the treatment effect, - γ^2 is the between-cluster variance.

The simulation process involves the following steps: 1. Randomly assign treatment (X_i) to each cluster with equal probability for control and treatment. 2. Generate cluster-level means (μ_i) by sampling from a log-normal distribution, where the log-scale mean is determined by $\alpha + \beta X_i$, and the log-scale variance is γ^2 . 3. Generate individual-level observations (Y_{ij}) within each cluster i by sampling from a Poisson distribution with mean μ_i .

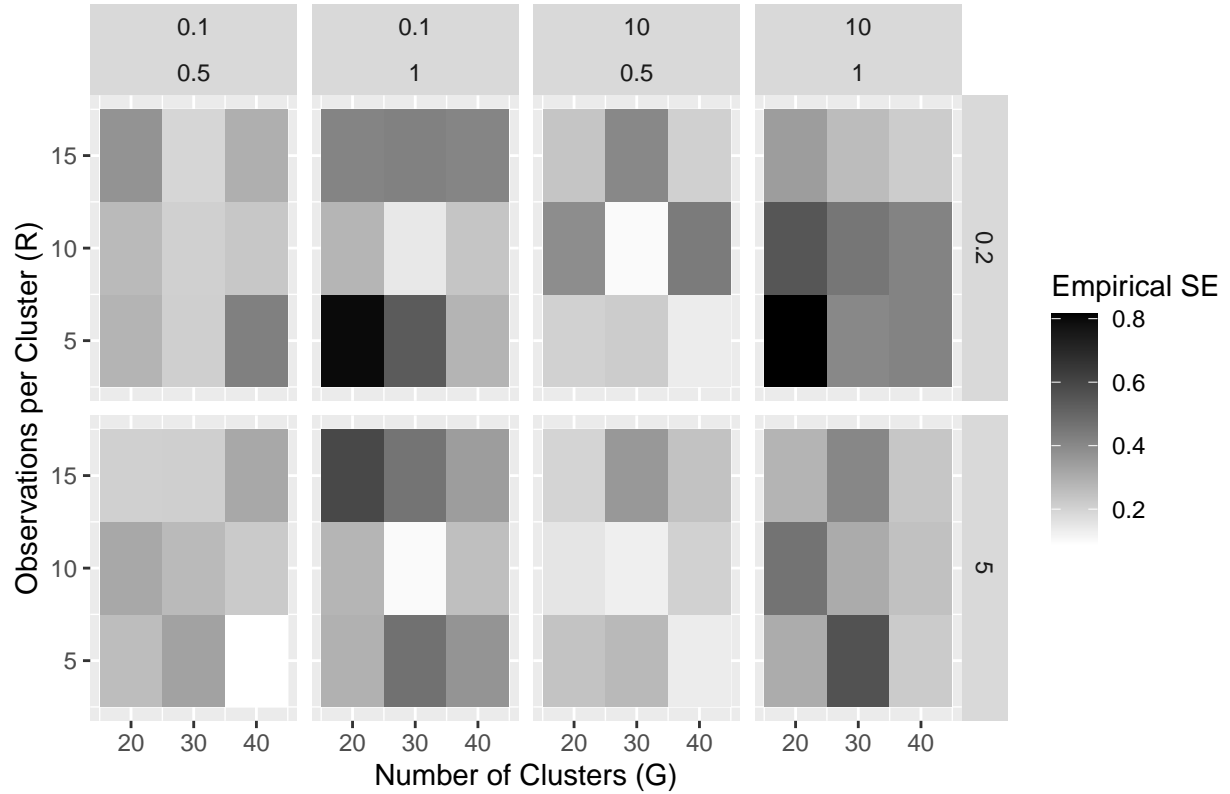
This data generation framework reflects the hierarchical structure of the model, where cluster-level effects (μ_i) account for variability between clusters, and individual-level counts (Y_{ij}) are conditionally independent given μ_i . The simulation allows for varying the number of clusters (G), the number of observations per cluster (R), and the variance parameter (γ^2) to evaluate their impact on performance metrics under the Poisson model. Below is the head of a sample generated data.

##	cluster	treatment	observation	Y
## 1	1	0	1	10
## 2	1	0	2	5
## 3	1	0	3	8
## 4	1	0	4	7
## 5	1	0	5	12
## 6	2	1	1	6

Analysis

We shall follow the similar method as in normal case using same metrics. The only difference is that we will use a generalized linear mixed-effects model (GLMM) to estimate . We will use a heat map to check how various values of , , and will affect the impact of G and R under no budget limit case. Then, we will use a similar grid search to find optimal G and R when budget is applied.

Empirical SE Across Varying G, R, Alpha, Beta, and Gamma2



Analysis of Empirical SE Trends when no budget limit

1. Effect of α (Baseline Outcome Level) The baseline outcome level (α) significantly influences the variability of the treatment effect estimates by determining the scale of the Poisson mean (μ_i). In the heatmap, smaller $\alpha = 0.2$ results in generally higher Empirical SE compared to larger $\alpha = 5$. This occurs because smaller α produces smaller Poisson means (μ_i), which in turn leads to higher relative variability in the observed counts (Y_{ij}). In this case, the Poisson variance (μ_i) is low, and the treatment effect estimates (β) become less stable, especially when R is small. Conversely, larger α stabilizes the Poisson outcomes, leading to lower Empirical SE across all combinations of G and R . This effect is more visible at smaller values of R , where the within-cluster variability has less opportunity to average out.

2. Effect of β (Treatment Effect) The treatment effect (β) determines the magnitude of the difference between the treatment and control groups. When β is small ($\beta = 0.1$), the treatment effect is weak, making it harder to distinguish between the two groups. This leads to higher Empirical SE overall, as the signal-to-noise ratio is lower. Increasing G and R still helps reduce Empirical SE, but the reduction is less pronounced due to the weak effect size. In contrast, when $\beta = 10$, the treatment effect is large, creating a clearer distinction between the groups. This results in lower Empirical SE, especially as G increases, because additional clusters better capture the strong effect. A large β strengthens the benefits of increasing G , making it an effective strategy for reducing variability.

3. Effect of γ^2 (Between-Cluster Variability) The between-cluster variability (γ^2) plays a critical role in determining the importance of increasing G . When $\gamma^2 = 0.5$, the variability between clusters is relatively small, so adding more clusters has a limited impact on reducing Empirical SE. In this scenario, increasing R (within-cluster observations) becomes more important, as it averages out the noise within clusters. However, when $\gamma^2 = 1$, the between-cluster variability is higher, and increasing G leads to a sharper reduction in Empirical SE. This is because additional clusters better capture the underlying variability across groups, improving the precision of the treatment effect estimates. The heatmap clearly shows that the effect of increasing G is more important when γ^2 is large, emphasizing its importance in high-variability scenarios.

Optimal G and R

Table 6: Grid Search Results for Poisson Model

G	R	Bias	Variance	MSE	Empirical_SE	Avg_SE
20	41	0.0156679	0.2118799	0.2121254	0.4603041	0.4209602
25	31	-0.0048983	0.1614696	0.1614936	0.4018328	0.3958932
30	24	-0.0326103	0.1415968	0.1426602	0.3762935	0.3563796
35	19	-0.0292017	0.1340816	0.1349343	0.3661715	0.3413565
40	16	0.0070301	0.0804751	0.0805245	0.2836813	0.3165637
45	13	-0.0065798	0.1282415	0.1282848	0.3581083	0.2947676
50	11	-0.0103399	0.0737290	0.0738360	0.2715309	0.2827843
55	9	-0.0136620	0.0909834	0.0911701	0.3016346	0.2698799
60	7	-0.0243157	0.0814050	0.0819963	0.2853156	0.2608067
65	6	-0.0028701	0.0678011	0.0678093	0.2603865	0.2508533
70	5	-0.0130272	0.0779686	0.0781383	0.2792286	0.2459935
75	4	-0.0053015	0.0839912	0.0840193	0.2898124	0.2397964
80	3	0.0147867	0.0561611	0.0563798	0.2369834	0.2321705
85	2	-0.0131577	0.0423342	0.0425074	0.2057529	0.2300681
90	2	-0.0062338	0.0655664	0.0656052	0.2560593	0.2225134

Table 7: Grid Search Results for Poisson Model

G	R	Bias	Variance	MSE	Empirical_SE	Avg_SE
20	176	-0.0361098	0.1767485	0.1780524	0.4204146	0.4339351
25	126	-0.0044631	0.1933911	0.1934110	0.4397626	0.3832580
30	92	-0.0125693	0.1434277	0.1435856	0.3787184	0.3559455
35	68	-0.0197991	0.1454663	0.1458583	0.3814004	0.3265272
40	51	0.0182980	0.1059060	0.1062408	0.3254320	0.3083718
45	37	0.0246345	0.1092497	0.1098565	0.3305293	0.2917505
50	26	-0.0060240	0.0820673	0.0821036	0.2864740	0.2783841
55	16	0.0145104	0.0832998	0.0835104	0.2886171	0.2662530

G	R	Bias	Variance	MSE	Empirical_SE	Avg_SE
60	9	0.0044308	0.0727657	0.0727854	0.2697512	0.2592747
65	2	-0.0704317	0.0801694	0.0851300	0.2831420	0.2565970

We can see that in the Poisson case, increase in c_1/c_2 ratio also results in reduced ratio of G/R .

Conclusion

Our project investigated how to design optimal cluster-randomized trials by analyzing the effects of varying the number of clusters (G) and observations per cluster (R) on estimation precision. We explored how data parameters (α , β , γ^2 , σ^2) and budget constraints influence performance metrics, particularly Empirical SE. Our findings show that increasing G is more effective when between-cluster variability (γ^2) is high, while increasing R is better for high within-cluster variability (σ^2). Larger baseline outcomes (α) and stronger treatment effects (β) improve precision and amplify the benefits of increasing G or R . Additionally, budget and cost ratios (c_1/c_2) play a crucial role, with high c_1/c_2 favoring R and low c_1/c_2 favoring G . We also extend our results to the Poisson case and achieve similar result.

Code Appendix:

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

library(tidyverse)
library(kableExtra)
library(RColorBrewer)
library(ggplot2)
library(dplyr)
library(tidyr)
library(Matrix)
library(knitr)
library(lme4)

generate_simulation <- function(G, R, alpha, beta, gamma2, sigma2) {
  # Generate treatment assignment for clusters
  X <- rbinom(G, 1, 0.5) # Randomly assign treatment (1) or control (0)

  # Generate cluster means
  mu <- rnorm(G, mean = alpha + beta * X, sd = sqrt(gamma2))

  # Generate individual observations
  Y <- vector("list", G)
  for (i in 1:G) {
    Y[[i]] <- rnorm(R, mean = mu[i], sd = sqrt(sigma2))
  }

  # Combine data into a dataframe
  data <- do.call(rbind, lapply(1:G, function(i) {
    data.frame(
      cluster = i,
      treatment = X[i],
      observation = 1:R,
      Y = Y[[i]]
    )
  })))

  return(data)
}

# Example usage
set.seed(123)
simulated_data <- generate_simulation(
  G = 10, R = 5,
  alpha = 1, beta = 0.5,
  gamma2 = 0.5, sigma2 = 1.0
)
head(simulated_data)
estimate_beta <- function(data) {
  model <- lmer(Y ~ treatment + (1 | cluster), data = data)
  summary_model <- summary(model)
  beta_hat <- summary_model$coefficients["treatment", "Estimate"]
  se <- summary_model$coefficients["treatment", "Std. Error"]
}
```

```

    return(list(beta_hat = beta_hat, se = se))
  }

evaluate_performance <- function(G, R, alpha, beta, gamma2, sigma2, n_sim = 100) {
  results <- replicate(n_sim, {
    data <- generate_simulation(G, R, alpha, beta, gamma2, sigma2)
    estimation <- estimate_beta(data)
    c(beta_hat = estimation$beta_hat, se = estimation$se)
  })

  beta_hats <- results["beta_hat", ]
  ses <- results["se", ]

  bias <- mean(beta_hats) - beta
  variance <- var(beta_hats)
  mse <- bias^2 + variance
  empirical_se <- sd(beta_hats)

  return(data.frame(
    G = G, R = R,
    gamma2 = gamma2, sigma2 = sigma2,
    Bias = bias, Variance = variance, MSE = mse,
    Empirical_SE = empirical_se, Avg_SE = mean(ses)
  ))
}

set.seed(123)
# Example usage
performance <- evaluate_performance(
  G = 10, R = 5,
  alpha = 1, beta = 0.5,
  gamma2 = 0.5, sigma2 = 1.0,
  n_sim = 100
)
kable(performance)

set.seed(123)
scenarios <- list(
  list(gamma2 = 10, sigma2 = 0.1), # Large gamma, small sigma
  list(gamma2 = 0.1, sigma2 = 10), # Small gamma, large sigma
  list(gamma2 = 1, sigma2 = 1)    # Gamma approximately equals sigma
)

# Generate results for all scenarios
results_all <- do.call(rbind, lapply(scenarios, function(scenario) {
  gamma2 <- scenario$gamma2
  sigma2 <- scenario$sigma2
  do.call(rbind, lapply(c(5, 20, 50), function(R) {
    evaluate_performance(G = 20, R = R, alpha = 1, beta = 0.5, gamma2 = gamma2, sigma2 = sigma2, n_sim = 100)
  })))
}))

kable(results_all, caption = "Performance Evaluation with fixed G")

```

```

set.seed(123)
scenarios <- list(
  list(gamma2 = 10, sigma2 = 0.1),
  list(gamma2 = 0.1, sigma2 = 10),
  list(gamma2 = 1, sigma2 = 1)
)

results_all <- do.call(rbind, lapply(scenarios, function(scenario) {
  gamma2 <- scenario$gamma2
  sigma2 <- scenario$sigma2

  do.call(rbind, lapply(c(20, 30, 50), function(G) {
    evaluate_performance(G = G, R = 20, alpha = 1, beta = 0.5, gamma2 = gamma2, sigma2 = sigma2, n_sim = 100)
  })))
}))

kable(results_all, caption = "Performance Evaluation with fixed R")

grid_search <- function(B, c1, c2, alpha, beta, gamma2, sigma2, n_sim = 100) {
  G_values <- seq(20, B / c1, by = 5)
  results <- lapply(G_values, function(G) {

    R <- floor((B - G * c1) / (G * c2) + 1)
    if (R > 1) {
      performance <- evaluate_performance(G, R, alpha, beta, gamma2, sigma2, n_sim)
      performance$G <- G
      performance$R <- R
      return(performance)
    }
    return(NULL)
  })

  results <- do.call(rbind, results)
  return(results)
}

set.seed(123)
B <- 1000
c1 <- 10
c2 <- 1
alpha <- 1
beta <- 0.5
gamma2 <- 10
sigma2 <- 0.1

results <- grid_search(B, c1, c2, alpha, beta, gamma2, sigma2, n_sim = 100)

optimal <- results[which.min(results$Empirical_SE), ]

ggplot(results, aes(x = G, y = R, fill = Empirical_SE)) +
  geom_tile() +
  geom_point(data = optimal, aes(x = G, y = R), color = "yellow", size = 3) +

```



```

scale_fill_gradient(low = "blue", high = "red") +
labs(title = "Case when Gamma greater than Sigma",
      x = "Number of Clusters (G)",
      y = "Samples per Cluster (R)",
      fill = "Empirical_SE ")

gamma2 <- 0.1
sigma2 <- 10
set.seed(123)
results <- grid_search(B, c1, c2, alpha, beta, gamma2, sigma2, n_sim = 100)

optimal <- results[which.min(results$Empirical_SE ), ]

ggplot(results, aes(x = G, y = R, fill = Empirical_SE )) +
  geom_tile() +
  geom_point(data = optimal, aes(x = G, y = R), color = "yellow", size = 3) +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(title = "Case when Gamma less than Sigma",
        x = "Number of Clusters (G)",
        y = "Samples per Cluster (R)",
        fill = "Empirical_SE ")

gamma2 <- 1
sigma2 <- 1
set.seed(123)
results <- grid_search(B, c1, c2, alpha, beta, gamma2, sigma2, n_sim = 100)

optimal <- results[which.min(results$Empirical_SE ), ]

ggplot(results, aes(x = G, y = R, fill = Empirical_SE )) +
  geom_tile() +
  geom_point(data = optimal, aes(x = G, y = R), color = "yellow", size = 3) +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(title = "Case when Gamma same as Sigma",
        x = "Number of Clusters (G)",
        y = "Samples per Cluster (R)",
        fill = "Empirical_SE ")

kable(results, caption = "Grid Search Results for Budget Allocation")
set.seed(123)
B <- 1000
c1 <- 15
c2 <- 0.2
alpha <- 1
beta <- 0.5
gamma2 <- 1
sigma2 <- 1
results <- grid_search(B, c1, c2, alpha, beta, gamma2, sigma2, n_sim = 100)

kable(results, caption = "Grid Search Results for Budget Allocation")

optimal <- results[which.min(results$Empirical_SE ), ]

```

```

ggplot(results, aes(x = G, y = R, fill = Empirical_SE )) +
  geom_tile() +
  geom_point(data = optimal, aes(x = G, y = R), color = "yellow", size = 3) +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(title = "Case when Gamma same as Sigma",
       x = "Number of Clusters (G)",
       y = "Samples per Cluster (R)",
       fill = "Empirical_SE ")

set.seed(123)
generate_simulation_poisson <- function(G, R, alpha, beta, gamma2) {
  # Generate treatment assignment for clusters
  X <- rbinom(G, 1, 0.5) # Randomly assign treatment (1) or control (0)

  # Generate cluster means
  log_mu <- rnorm(G, mean = alpha + beta * X, sd = sqrt(gamma2))
  mu <- exp(log_mu) # Back-transform to original scale

  # Generate individual observations
  data <- do.call(rbind, lapply(1:G, function(i) {
    Y <- rpois(R, lambda = mu[i])
    data.frame(
      cluster = i,
      treatment = X[i],
      observation = 1:R,
      Y = Y
    )
  })))

  return(data)
}

head(generate_simulation_poisson(20,5,1,0.5,1))
estimate_beta_poisson <- function(data) {
  library(lme4)
  model <- glmer(Y ~ treatment + (1 | cluster), family = poisson, data = data)
  summary_model <- summary(model)
  beta_hat <- summary_model$coefficients["treatment", "Estimate"]
  se <- summary_model$coefficients["treatment", "Std. Error"]
  return(list(beta_hat = beta_hat, se = se))
}

evaluate_performance_poisson <- function(G, R, alpha, beta, gamma2, n_sim = 100) {
  results <- replicate(n_sim, {
    data <- generate_simulation_poisson(G, R, alpha, beta, gamma2)
    estimation <- estimate_beta_poisson(data)
    c(beta_hat = estimation$beta_hat, se = estimation$se)
  })

  beta_hats <- results["beta_hat", ]
  ses <- results["se", ]

  bias <- mean(beta_hats) - beta

```

```

variance <- var(beta_hats)
mse <- bias^2 + variance
empirical_se <- sd(beta_hats)

return(data.frame(
  G = G, R = R,
  Bias = bias, Variance = variance, MSE = mse,
  Empirical_SE = empirical_se, Avg_SE = mean(ses)
))
}

# Simulate data under varying alpha, beta, and gamma2 without budget constraints
simulate_empirical_se <- function(alpha, beta, gamma2, G_values, R_values, n_sim = 5) {
  results <- expand.grid(G = G_values, R = R_values) %>%
    group_by(G, R) %>%
    summarise(
      Empirical_SE = sd(replicate(n_sim, {
        data <- generate_simulation_poisson(G = G, R = R, alpha = alpha, beta = beta, gamma2 = gamma2)
        estimation <- estimate_beta_poisson(data)
        estimation$beta_hat
      })))
    )
  results$alpha <- alpha
  results$beta <- beta
  results$gamma2 <- gamma2
  return(results)
}

alpha_values <- c(0.2, 5)
beta_values <- c(0.1, 10)
gamma2_values <- c(0.5, 1)
G_values <- c(20, 30, 40)
R_values <- c(5, 10, 15)

# Collect results for all parameter combinations
empirical_se_results <- do.call(rbind, lapply(alpha_values, function(alpha) {
  do.call(rbind, lapply(beta_values, function(beta) {
    do.call(rbind, lapply(gamma2_values, function(gamma2) {
      simulate_empirical_se(alpha, beta, gamma2, G_values, R_values)
    })))
  })))
}))

library(ggplot2)

ggplot(empirical_se_results, aes(x = G, y = R, fill = Empirical_SE)) +
  geom_tile() +
  facet_grid(alpha ~ beta + gamma2) +
  scale_fill_gradient(low = "white", high = "black") +
  labs(
    title = "Empirical SE Across Varying G, R, Alpha, Beta, and Gamma2",
    x = "Number of Clusters (G)",
    y = "Observations per Cluster (R)",

```

```

    fill = "Empirical SE"
  )

grid_search_poisson <- function(B, c1, c2, alpha, beta, gamma2) {
  G_values <- seq(20, B / c1, by = 5) # G must satisfy B > G * c1
  results <- do.call(rbind, lapply(G_values, function(G) {
    R <- floor((B - G * c1) / (G * c2) + 1)
    if (R > 1) { # Ensure R is valid
      performance <- evaluate_performance_poisson(G, R, alpha, beta, gamma2)
      performance$G <- G
      performance$R <- R
      return(performance)
    }
    return(NULL)
  }))
  return(results)
}

set.seed(123)
# Parameters
B <- 1000 # Total budget
c1 <- 10  # Cost of the first observation in a cluster
c2 <- 1   # Cost of additional observations
alpha <- 1
beta <- 0.5
gamma2 <- 1

# Run grid search
results_poisson <- grid_search_poisson(B, c1, c2, alpha, beta, gamma2)

kable(results_poisson, caption = "Grid Search Results for Poisson Model")

set.seed(123)
# Parameters
B <- 1000
c1 <- 15
c2 <- 0.2
alpha <- 1
beta <- 0.5
gamma2 <- 1

# Run grid search
results_poisson <- grid_search_poisson(B, c1, c2, alpha, beta, gamma2)

kable(results_poisson, caption = "Grid Search Results for Poisson Model")

```