Project3-Simulation

Yunan Chen

2024-12-02

Introducation

Cluster randomized trials are widely used in public health and clinical research, where clusters of units are randomized to either control or treatment groups to estimate treatment effects. For example, in a study evaluating the effectiveness of a new hand hygiene protocol, hospitals were randomized to either implement the protocol or continue standard practices. Within each hospital, individual patient outcomes, such as rates of hospital-acquired infections, were measured to assess the impact of the intervention (Clancy et al., 2020). In another study, individuals were treated as clusters, and repeated measurements, such as daily blood glucose levels, were collected to assess the effect of a dietary intervention on long-term glycemic control (Karimian et al., 2023).

In our study, we consider clusters as individuals and the number of observations per cluster as repeated measurements of these individuals. Designing an efficient cluster randomized trial requires balancing the number of clusters, the number of repeated measurements per cluster, and the associated costs while maximizing statistical precision under a fixed budget. A critical design challenge for cluster randomized trials lies in allocating a fixed budget to maximize the precision of treatment effect estimates. This challenge is further complicated by uncertainties in intracluster correlation coefficients, which measure within-cluster similarity. While independent observations are generally preferred for statistical power, budget constraints often necessitate strategic trade-offs between the number of clusters and the number of measurements per cluster. In some cases, obtaining additional measurements within the same cluster or subject can be significantly less expensive than recruiting new clusters or subjects. For example, with a fixed budget, sampling a smaller number of subjects with repeated measurements may yield comparable or even improved precision for estimating treatment effects by reducing residual variance.

This project, in collaboration with Dr. Zhijin Wu from the Biostatistics Department, aims to develop a simulation study to identify optimal cluster randomized trial designs under budgetary constraints. Using simulation studies guided by the aim, data structure, estimand, methods, and performance (ADEMP) framework, the study systematically explores trade-offs between cluster-level and within-cluster observations under varying cost structures and statistical assumptions. By evaluating resource allocation strategies, the project provides practical recommendations for designing cost-effective and statistically robust cluster randomized trials that maximize the precision of treatment effect estimates. Additionally, extending the simulations to outcomes modeled with a Poisson distribution will offer insights into how hierarchical modeling assumptions impact design recommendations, contributing to advancing the methodology for resource-efficient experimental designs in cluster randomized trials.

Method

To achieve the aims of the project, we are employing the ADEMP framework, a structured approach for designing and reporting simulation studies, ensuring clarity, reproducibility, and thorough evaluation of statistical methods. The framework includes five key elements: Aims, which specify the study's objectives, such

as evaluating bias, precision, or power; Data-generating mechanisms, which define how data are simulated, including underlying models and parameter values; Estimands, identifying the quantities or parameters of interest being assessed; Methods, describing the statistical techniques or models applied to estimate these quantities; and Performance measures, which evaluate the methods' effectiveness using metrics like bias, mean squared error, variance or power. This systematic approach guides the design and analysis of simulation studies to achieve robust and meaningful insights. By adopting this structured approach, the framework ensures methodological rigor, enhances reproducibility, and provides a comprehensive evaluation of statistical methods within the context of simulation-based research.

Simulation Framework

The simulation study adheres to the ADEMP framework to evaluate optimal designs for cluster randomized trials under budget constraints.

Aims:

To evaluate the impacts (1) of the optimal allocation of a fixed budget across the number of clusters (G) and the number of measurements per cluster (R) in a cluster randomized trial, focusing on minimizing the variance of the estimated treatment effect (β) while balancing trade-offs between within-cluster and between-cluster variability; (2) of underlying data-generating parameters, such as β and σ^2 for Normal outcomes and β and γ^2 for Poisson outcomes, as well as relative cost ratios (c_1/c_2), on the precision and power of the treatment effect estimate; and (3) of differences in performance under Normal and Poisson data-generating mechanisms, investigating how distributional assumptions influence the results and optimal design strategies.

Data-Generating Mechanisms:

- Normal Outcomes: For observation $j = 1, \cdots, R$ within cluster $i = 1, \cdots, G$, generate the data under the hierarchical structure:

$$\begin{aligned} \mu_{i0} &= \alpha + \beta X_i \quad \text{(fixed effects, where } X_i = 0 \text{ for control and } X_i = 1 \text{ for treatment)}, \\ \mu_i \mid \epsilon_i &= \mu_{i0} + \epsilon_i, \quad \epsilon_i \sim N(0, \gamma^2), \\ Y_{ij} \mid \mu_i &= \mu_i + e_{ij}, \quad e_{ij} \sim N(0, \sigma^2). \end{aligned}$$

- Poisson Outcomes: For observation $j = 1, \cdots, R$ within cluster $i = 1, \cdots, G$, let:

$$\begin{aligned} &\log(\mu_i) \sim N(\alpha + \beta X_i, \gamma^2), \quad \text{where } X_i = 0 \text{ for control and } X_i = 1 \text{ for treatment}, \\ &Y_{ij} \mid \mu_i \sim \text{Poisson}(\mu_i), \quad \text{for } j = 1, \ldots, R \end{aligned}$$

Estimand: The primary estimand is the treatment effect, β, which quantifies the difference in outcomes between the treatment ($X_i = 1$) and control ($X_i = 0$) groups. For Normal outcomes, β represents the average difference in means. For Poisson outcomes, β corresponds to the log-risk ratio of the outcome rates.

Methods:

For each simulation, the generated data set (X, Y) under normal or Poisson data-generating mechanisms is analyzed using a GLM model, and the estimate β is then extracted from the model result.

Performance Measures:

We will assess the performance of the study design using two key metrics: variance and power. Variance measures the variability of $\hat{\beta}$ across simulations, providing insights into how design parameters, such as the number of clusters and measurements per cluster, impact the precision of the treatment effect estimate. Power quantifies the ability to detect significant treatment effects by calculating the proportion of simulations where $\hat{\beta}$ is significantly different from zero. This evaluation focuses on the influence of cost ratios and variability parameters on the statistical power of the study design.

Varying parameters

The optimization process involved two major steps. First, we fixed the key parameters of the distribution: $\alpha = 2, \beta = 1.5, \sigma^2 = 1, \gamma^2 = 1$ These values were chosen because they represent moderate, plausible magnitudes for intercepts, effect sizes, and variances in typical clustered designs, allowing us to explore the study design under realistic and interpretable baseline conditions. With these baseline values and a fixed budget of 2000, we systematically varied the relative cost $\frac{c_1}{c_2}$ across 2, 5, 10, and 20, and the number of clusters over a sequence from 10 to 50 in increments of 5. To compute the cost per observation c_2, we used $c_2 = \frac{c_1}{\text{relative cost}}$. The number of observations per cluster was then calculated as number of Observations per Cluster $= \frac{\text{Budget} - n_{clusters} \cdot c_1}{n_{clusters} \cdot c_2}$. For example, under a fixed budget of 2000, if the relative cost $\frac{}{c_1}c_2$ is 2 and there are 20 clusters, the number of observations per cluster would be 98. However, if the relative cost increases to 10 while keeping the number of clusters at 20, the number of observations per cluster decreases to 90. This demonstrates that higher relative cluster costs or an increase in the number of clusters reduces the number of observations per cluster, as the budget must be distributed accordingly. The goal of this step was to identify the optimal combination of the number of clusters and the number of observations per cluster that minimizes the variance of the estimated treatment effect β.

After determining the optimal combination of the number of clusters and the number of observations per cluster, we fixed these values and conducted additional simulations to assess how varying the distribution parameters influences the optimal study design. Specifically, we varied the residual variance σ^2 over a sequence from 0.01 to 10 with 10 equally spaced values, and the treatment effect β across 0.05, 0.5, and 1.5. Changes in σ^2 directly affect the intra-class correlation (ICC), which quantifies the proportion of the total variance attributable to clustering effects, given by ICC $= \frac{\gamma^2}{\gamma^2 + \sigma^2} ICC$. Under a fixed number of clusters and observations per cluster, increasing σ^2 reduces the ICC, indicating weaker clustering effects. This, in turn, increases the within-cluster variation relative to the total variation, which can result in greater variability of the β estimate. As σ^2 increases, the residual noise dominates, making it harder to detect the treatment effect precisely and potentially widening confidence intervals for β. Conversely, lower values of σ^2 strengthen clustering effects, reducing the variability of β and improving the precision of the estimate. The parameter α, representing the intercept, was not varied because it does not influence the variance or clustering structure of the data. Instead, α shifts the overall distribution of outcomes without altering the relationships between the variables or the precision of the estimates, and therefore has no direct impact on the performance metrics of the study design. This step allowed us to evaluate not only the robustness of the optimal design but also how the degree of residual variance impacts the precision of β under different data generation conditions. By combining these two steps, we were able to comprehensively evaluate and refine the study design to achieve a balance between cost efficiency and statistical precision.

Results

Optimal Q and R

`Table 1` summarizes the variance of the treatment effect estimate ($\hat{\beta}$) and the corresponding number of measurements per cluster (R) across various combinations of the number of clusters (G) and relative cost ratios (c_1/c_2) for both Normal and Poisson data-generating mechanisms. Each row represents a specific number of clusters (G), while the columns provide the corresponding R and the variance of $\hat{\beta}$ for cost ratios $c_1/c_2 = 2, 5, 10, 20$, under a fixed budget of 2000. The highlighted cells identify the optimal combinations of G and R that achieve the lowest variance of $\hat{\beta}$, reflecting the most efficient allocation of resources within the given constraints.

Normal Distribution

Under the Normal distribution, `Figure 1` visualizes the relationship between the variance of the treatment effect estimate ($\hat{\beta}$) and the number of clusters (G), stratified by relative cost ratios ($c_1/c_2 = 2, 5, 10, 20$). Each

Table 1: Simulation Results Vary by Number of Clusters and Relative Costs

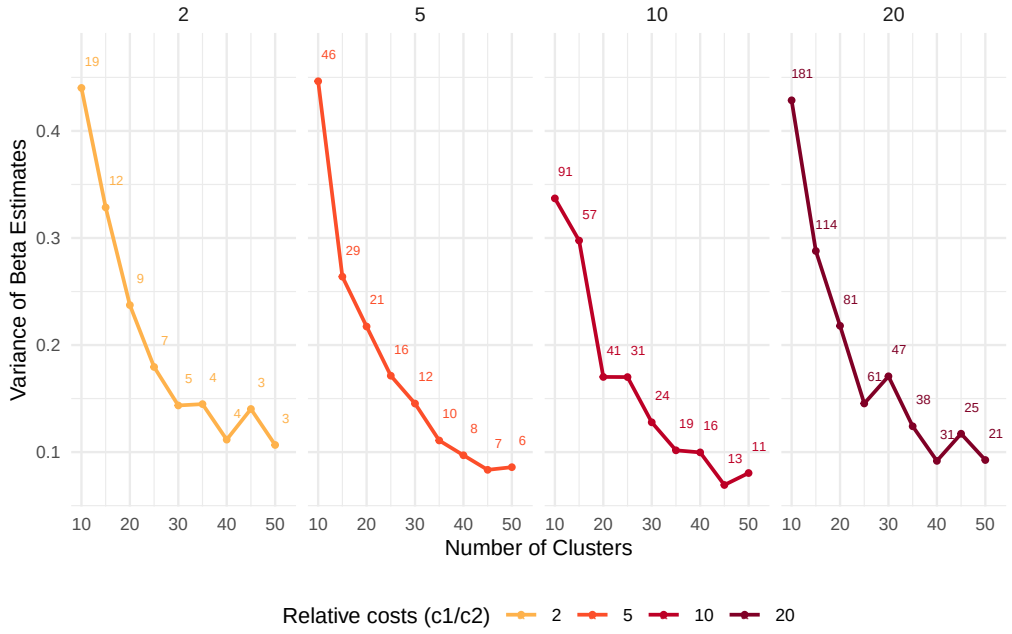
	c1/c2=2		c1/c2=5		c1/c2=10		c1/c2=20	
G	R	Var(beta_hat)	R	Var(beta_hat)	R	Var(beta_hat)	R	Var(beta_hat)
Normal								
10	19	0.44	46	0.45	91	0.34	181	0.43
15	12	0.33	29	0.26	57	0.30	114	0.29
20	9	0.24	21	0.22	41	0.17	81	0.22
25	7	0.18	16	0.17	31	0.17	61	0.15
30	5	0.14	12	0.15	24	0.13	47	0.17
35	4	0.14	10	0.11	19	0.10	38	0.12
40	4	0.11	8	0.10	16	0.10	31	0.09
45	3	0.14	7	0.08	13	0.07	25	0.12
50	3	0.11	6	0.09	11	0.08	21	0.09
Poisson								
10	19	0.29	46	0.44	91	0.42	181	0.30
15	12	0.31	29	0.27	57	0.31	114	0.29
20	9	0.14	21	0.17	41	0.20	81	0.19
25	7	0.16	16	0.18	31	0.16	61	0.20
30	5	0.14	12	0.17	24	0.14	47	0.12
35	4	0.13	10	0.12	19	0.13	38	0.12
40	4	0.08	8	0.12	16	0.07	31	0.11
45	3	0.07	7	0.09	13	0.07	25	0.10
50	3	0.08	6	0.08	11	0.09	21	0.07

line represents a specific cost ratio, and the labels indicate the corresponding number of measurements per cluster (R) at different cluster sizes. The variance of $\hat{\beta}$ decreases as the number of clusters increases, a trend consistent with the results presented in `Table 1`. However, the decrease follows a pattern of diminishing returns, with minimal reductions in variance observed when the number of clusters exceeds approximately $G = 40$.

`Figure 1` also highlights the trade-offs between within-cluster measurements (R) and the number of clusters (G) at different cost ratios. For lower cost ratios ($c_1/c_2 = 2, 5$), more resources are allocated to within-cluster measurements, resulting in steeper reductions in variance at smaller G. The optimal configurations in these scenarios, such as $G = 50, R = 3$ for $c_1/c_2 = 2$ and $G = 45, R = 7$ for $c_1/c_2 = 5$, align closely with the results in Table 1, achieving variances of 0.11 and 0.08, respectively. In contrast, higher cost ratios ($c_1/c_2 = 10, 20$) prioritize increasing the number of clusters, leading to more gradual reductions in variance. Optimal configurations in these cases, such as $G = 45, R = 13$ or $G = 50, R = 11$, yield variances as low as 0.07.

Together, `Figure 1` and `Table 1` demonstrate that while increasing the number of clusters consistently reduces the variance of $\hat{\beta}$, the choice of cost ratio plays a critical role in determining the balance between G and R. This relationship underscores the importance of strategic resource allocation to achieve optimal precision under budget constraints in cluster randomized trials.

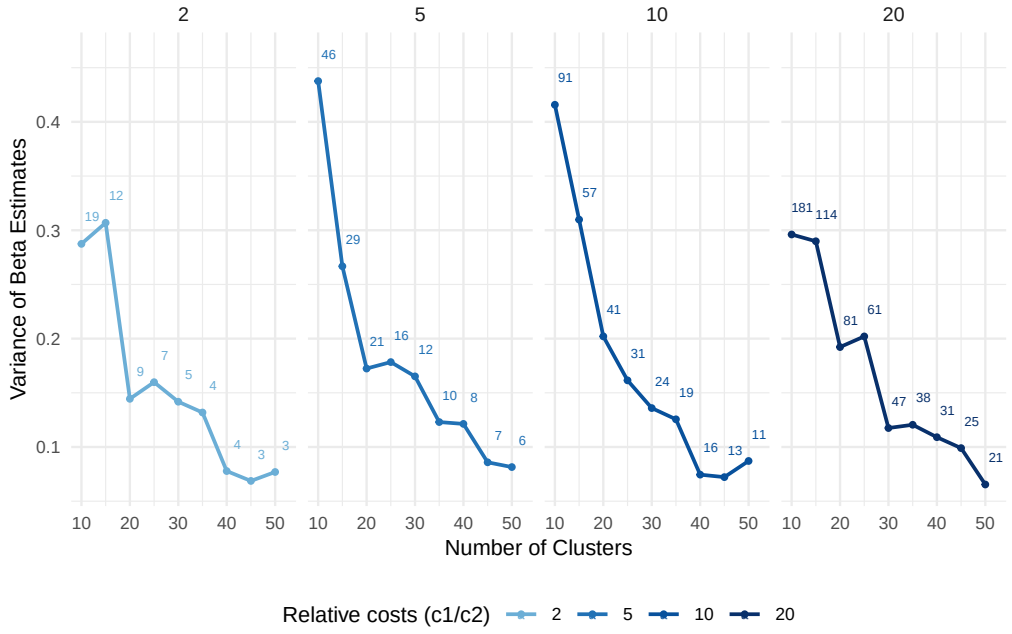
Figure 1: Variance of Beta Estimates vs. Number of Clusters



Poisson

Under the Poisson distribution, the variance of β_{est} follows a similar trend but is generally higher compared to the Normal distribution, particularly for smaller G. For instance, at G=10 and $\frac{c_1}{c_2}$=2, the variance is 0.29 under Poisson compared to 0.44 under Normal. This reflects the additional variability associated with Poisson-distributed outcomes. As G increases, the variance decreases, and by G=40 or higher, the differences between the two distributions narrow. Lower cost ratios again lead to smaller variances due to higher R. The plot for the Poisson distribution confirms these patterns, showing steeper reductions in variance at smaller G values and stabilization as G increases.

Figure 2: Variance of Beta Estimates vs. Number of Clusters



In summary, the results demonstrate that increasing G reduces the variance of β_{est} under both distributions, with more pronounced improvements at smaller G values. Higher cost ratios ($\frac{c_1}{c_2}$=10 or 20) result in higher variances due to fewer observations per cluster. The variance is consistently higher for the Poisson distribution at smaller G, but this difference diminishes as G increases, reflecting the stabilizing effect of larger sample sizes. These findings highlight the importance of optimizing the number of clusters, observations per cluster, and cost ratios to ensure precise and efficient study designs under varying distributions.

Varying data Generation

This table summarizes simulation results for the Normal and Poisson distributions under varying cost ratios (c_1/c_2), variances (σ^2 and γ^2), and treatment effect sizes (β). As the cost ratio increases, the number of clusters (G) decreases, and the number of observations per cluster (R) increases, reflecting the allocation of resources between clusters and individuals. For the Normal distribution, the variability of the beta estimate ($\text{Var}(\beta_{\text{est}})$) remains relatively stable, and power is generally higher, especially at lower variances ($\sigma^2 = 0.1$). In contrast, the Poisson distribution shows higher variability in beta estimates and lower power overall, particularly at larger variances ($\sigma^2 = 10$).

Increasing variance (σ^2) negatively impacts both distributions by increasing $\text{Var}(\beta_{\text{est}})$ and reducing power. However, the Poisson distribution is more sensitive to variance, leading to significant reductions in power for smaller treatment effects ($\beta = 0.05$). Larger treatment effects ($\beta = 1.50$) consistently improve power and reduce variability across both distributions, though the Normal distribution outperforms Poisson in terms of stability and power.

The balance between clusters and observations is crucial. For the Normal distribution, increasing the cost ratio (c_1/c_2) and allocating more observations per cluster tends to improve power and stabilize $\text{Var}(\beta_{\text{est}})$. Conversely, for the Poisson distribution, increasing the number of clusters (G) by reducing the cost ratio might help mitigate the higher variability observed with larger variances. This highlights the importance of careful design considerations in resource allocation, especially when dealing with high variance or smaller treatment effects. Overall, the Normal distribution demonstrates better performance in terms of lower variability and higher power, making it a preferable choice when its assumptions are met.

Normal

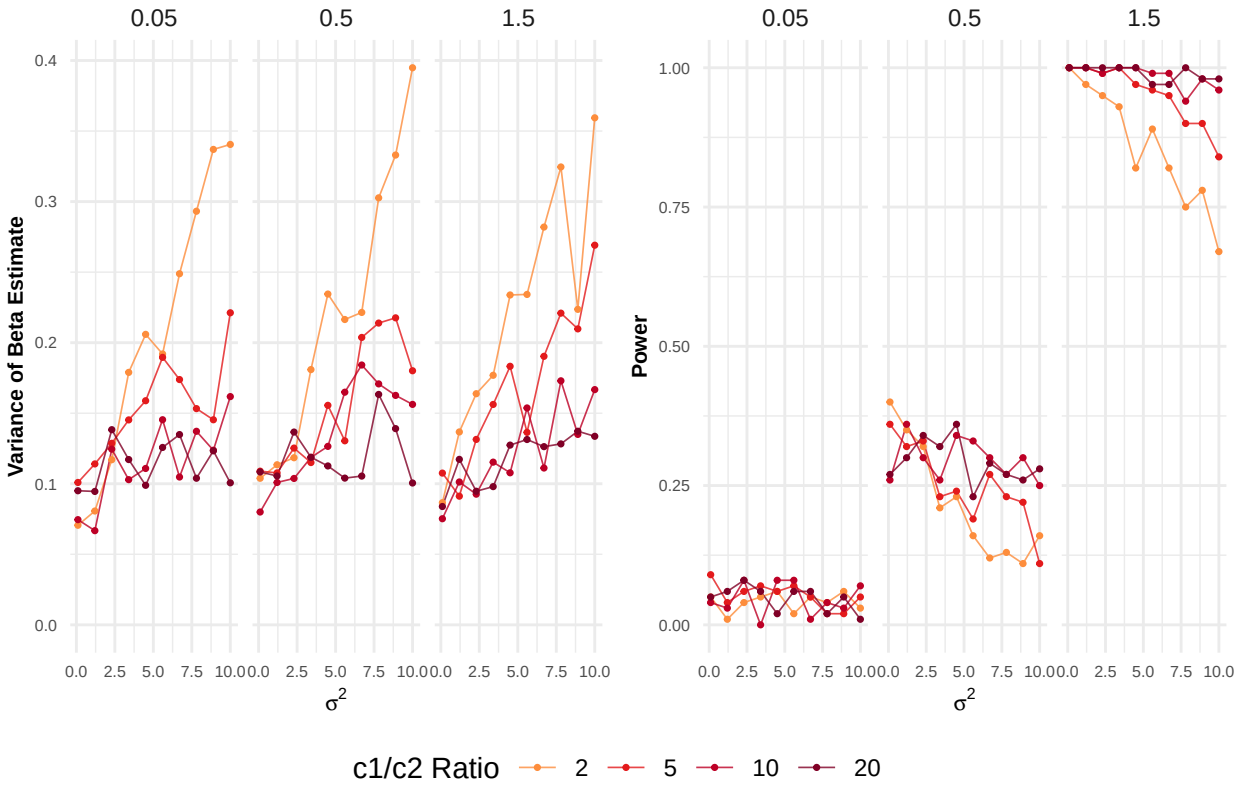
This plot illustrates the impact of residual variance (σ^2) on the variance of β_{est} and power under the Normal distribution, stratified by three treatment effect sizes (β =0.05,0.5,1.5) and four relative cost ratios ($\frac{c_1}{c_2}$=2,5,10,20). In the left panel, the variance of β_{est} increases as σ^2 grows, particularly beyond σ^2=5. This trend reflects the additional variability introduced by larger residuals, which reduces the precision of the β estimate. Lower cost ratios ($\frac{c_1}{c_2}$=2) exhibit slightly higher variances for smaller σ^2, while higher cost ratios ($\frac{c_1}{c_2}$=10 or 20) stabilize the variance as σ^2 increases by spreading the variability across more clusters. Larger treatment effects (β=1.5) consistently result in lower variances compared to smaller treatment effects (β=0.05), demonstrating that larger effect sizes are easier to estimate precisely, even in the presence of variability.

In the right panel, power decreases with increasing σ^2, particularly for smaller treatment effects (β=0.05), where detecting the effect becomes increasingly challenging as residual variability grows. Lower cost ratios ($\frac{c_1}{c_2}$ =2) achieve slightly higher power for small σ^2, as they allocate more resources to within-cluster observations, improving the ability to detect effects. In contrast, higher cost ratios ($\frac{c_1}{c_2}$=10 or 20) prioritize increasing the number of clusters, which can reduce power due to fewer observations per cluster. For large treatment effects (β=1.5), power is relatively robust to changes in σ^2 and cost ratios, reflecting the strong signal associated with larger effects. These results highlight the trade-offs between cost allocation, residual variability, and treatment effect size in study design, with smaller treatment effects being more sensitive to increased variability and cost distribution strategies.

Table 2: Simulation Results Vary by Variance and Treatment Effect

			Normal			Poisson		
c1/c2	sigma^2 (gamma^2)	**beta**	G (R)	Var(beta_hat)	power	G (R)	Var(beta_hat)	power
2	0.1	0.05	50 (3)	0.07	0.05	45 (3)	0.01	0.11
2	0.1	0.50	50 (3)	0.10	0.40	45 (3)	0.01	0.99
2	0.1	1.50	50 (3)	0.09	1.00	45 (3)	0.01	1.00
2	4.5	0.05	50 (3)	0.21	0.06	45 (3)	0.58	0.12
2	4.5	0.50	50 (3)	0.23	0.23	45 (3)	0.46	0.17
2	4.5	1.50	50 (3)	0.23	0.82	45 (3)	0.49	0.60
2	10.0	0.05	50 (3)	0.34	0.03	45 (3)	0.75	0.04
2	10.0	0.50	50 (3)	0.39	0.16	45 (3)	0.78	0.06
2	10.0	1.50	50 (3)	0.36	0.67	45 (3)	1.13	0.35
5	0.1	0.05	45 (7)	0.10	0.09	50 (6)	0.01	0.09
5	0.1	0.50	45 (7)	0.11	0.36	50 (6)	0.01	1.00
5	0.1	1.50	45 (7)	0.11	1.00	50 (6)	0.01	1.00
5	4.5	0.05	45 (7)	0.16	0.06	50 (6)	0.37	0.07
5	4.5	0.50	45 (7)	0.16	0.24	50 (6)	0.33	0.14
5	4.5	1.50	45 (7)	0.18	0.97	50 (6)	0.41	0.71
5	10.0	0.05	45 (7)	0.22	0.05	50 (6)	0.73	0.06
5	10.0	0.50	45 (7)	0.18	0.11	50 (6)	0.82	0.13
5	10.0	1.50	45 (7)	0.27	0.84	50 (6)	0.90	0.37
10	0.1	0.05	45 (13)	0.07	0.04	45 (13)	0.01	0.12
10	0.1	0.50	45 (13)	0.08	0.26	45 (13)	0.01	1.00
10	0.1	1.50	45 (13)	0.08	1.00	45 (13)	0.01	1.00
10	4.5	0.05	45 (13)	0.11	0.08	45 (13)	0.47	0.08
10	4.5	0.50	45 (13)	0.13	0.34	45 (13)	0.42	0.11
10	4.5	1.50	45 (13)	0.11	1.00	45 (13)	0.36	0.69
10	10.0	0.05	45 (13)	0.16	0.07	45 (13)	1.03	0.07
10	10.0	0.50	45 (13)	0.16	0.25	45 (13)	0.86	0.06
10	10.0	1.50	45 (13)	0.17	0.96	45 (13)	0.86	0.35
20	0.1	0.05	40 (31)	0.10	0.05	50 (21)	0.01	0.11
20	0.1	0.50	40 (31)	0.11	0.27	50 (21)	0.01	1.00
20	0.1	1.50	40 (31)	0.08	1.00	50 (21)	0.01	1.00
20	4.5	0.05	40 (31)	0.10	0.02	50 (21)	0.32	0.06
20	4.5	0.50	40 (31)	0.11	0.36	50 (21)	0.44	0.14
20	4.5	1.50	40 (31)	0.13	1.00	50 (21)	0.23	0.76
20	10.0	0.05	40 (31)	0.10	0.01	50 (21)	0.92	0.07
20	10.0	0.50	40 (31)	0.10	0.28	50 (21)	0.52	0.07
20	10.0	1.50	40 (31)	0.13	0.98	50 (21)	0.95	0.48

Figure 3: Variance of Beta Estimate and Power Across c1/c2 Ratios

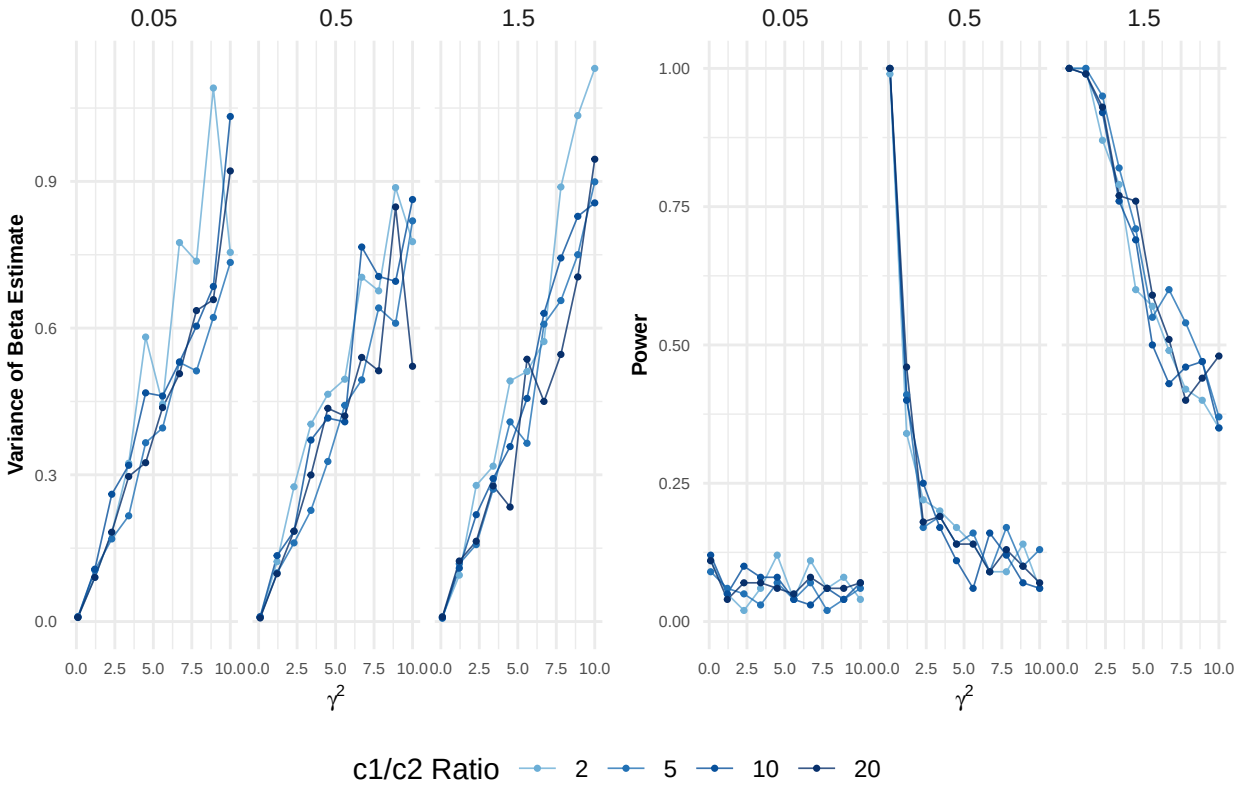


Poisson

This plot examines the variance of β_{est} (left panel) and power (right panel) as functions of γ^2 under the Poisson distribution, stratified by treatment effect sizes ($\beta = 0.5, 1.5, 3$) and relative cost ratios ($c_1/c_2 = 2, 5, 10, 20$). The variance of β_{est} increases consistently with γ^2 across all treatment effect sizes and cost ratios. Higher γ^2, representing greater between-cluster variability, adds uncertainty to β_{est}, with the most pronounced increases observed beyond $\gamma^2 = 5$. Lower cost ratios ($c_1/c_2 = 2$) result in slightly higher variances due to a larger number of within-cluster observations, amplifying the effects of between-cluster heterogeneity. Higher cost ratios ($c_1/c_2 = 10$ or 20), which prioritize more clusters, mitigate this effect by distributing variability across more clusters. Larger treatment effects ($\beta = 3$) generally show higher variances compared to smaller effects, as larger effects are more influenced by between-cluster heterogeneity.

The power to detect treatment effects decreases as γ^2 increases, particularly for smaller treatment effects ($\beta = 0.5$). Power is highest for smaller γ^2 values and larger treatment effects ($\beta = 3$), which are less sensitive to increases in between-cluster heterogeneity. Lower cost ratios ($c_1/c_2 = 2$) achieve slightly higher power for smaller γ^2 because they allocate more observations per cluster, improving the ability to detect small effects. Conversely, higher cost ratios ($c_1/c_2 = 10$ or 20) lead to lower power for small treatment effects as γ^2 grows, due to fewer within-cluster observations. These results highlight the trade-off between cost allocation strategies and the effects of between-cluster variability on precision and power, emphasizing the need to carefully account for γ^2 in study design.

Figure 4: Variance of Beta Estimate and Power Across c1/c2 Ratios



Comparison

The effects of increasing σ^2 in the Normal distribution and γ^2 in the Poisson distribution on the variance of β_{est} share several similarities but operate through distinct mechanisms. In both cases, increasing these parameters leads to higher variance of β_{est}, as they represent measures of variability in the data. The impact

of both parameters becomes more pronounced as their values grow, with steeper increases in variance observed at higher levels of σ^2 and γ^2. Additionally, the effects of both parameters are influenced by study design features, such as the number of clusters and the allocation of observations per cluster, which can mitigate their impact.

However, key differences exist between the two. In the Normal distribution, σ^2 represents residual variability at the individual level, affecting precision by amplifying within-cluster noise. In contrast, γ^2 in the Poisson distribution captures between-cluster variability in log-scale means, influencing the heterogeneity of cluster-specific rates and indirectly affecting within-cluster variation. The effect of σ^2 is more sensitive to the number of observations per cluster (R), as larger within-cluster samples reduce residual noise. In contrast, γ^2 primarily depends on the number of clusters (G), as adding clusters distributes variability across a larger sample. Moreover, σ^2 operates directly on the outcome's original scale, while γ^2 acts on the log-scale, amplifying differences between cluster-level means. These distinctions highlight how variability parameters uniquely influence the precision of β_{est} depending on the distributional assumptions and emphasize the importance of tailored strategies for study design optimization.

Discussion

Limitation

This project has several limitations. First, the number of simulations conducted for each scenario was limited to 100 due to computational constraints. While this choice was made to balance running time and feasibility, it may not fully capture the variability in the estimates, potentially leading to less stable conclusions about the optimal combinations of clusters (G) and measurements per cluster (R). Increasing the number of simulations would provide more robust and reliable results, albeit at the cost of significantly greater computational resources.

Second, the sequential approach used to determine the optimal design parameters introduces a limitation. The number of clusters (G) and relative cost ratios (c_1/c_2) were varied first to identify the optimal combination of G and R, and then the variance components (σ^2, γ^2) and treatment effects (β) were varied while keeping the optimal G and R fixed. This approach does not allow for the simultaneous variation of G, R, variance components, and treatment effects. As a result, potential interactions between these parameters could not be fully explored, which may limit the generalizability of the findings. Future work should consider varying all parameters together to better capture their joint effects on study outcomes.

Finally, the exploration of intra-class correlation (ICC) under the Normal distribution focused on varying σ^2 while keeping other parameters fixed. Although this approach provided insights into how ICC impacts the variance of $\hat{\beta}$, γ^2 was not varied. Both σ^2 and γ^2 contribute to ICC, but their individual impacts on the variance of $\hat{\beta}$ may differ. By not varying γ^2, the analysis may overlook important differences in how these parameters affect ICC and the resulting variance of $\hat{\beta}$. Future studies should address this by simultaneously varying σ^2 and γ^2 to provide a more comprehensive understanding of their respective roles in influencing ICC and study precision.

References

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)
library(tidyverse)
library(lmerTest)
library(lme4)
library(dplyr)
library(ggplot2)
library(ggpubr)
library(gt)
library(RColorBrewer)
# Simulate data and performance matrix under Normal distribution
sim_normal <- function(n_clusters, B, c1, c1_c2_ratio, alpha, beta, gamma2, sigma2, n_sim, alpha_level =
  set.seed(2550)
  # Calculate c2 from the ratio
  c2 <- c1 / c1_c2_ratio

  # Calculate the number of observations per cluster
  n_obs_per_cluster <- floor((B - n_clusters * c1) / (n_clusters * c2)) + 1

  # Assign clusters to treatment (X = 1) or control (X = 0)
  cluster_treatment <- rep(c(0, 1), n_clusters / 2)

  # Initialize data frame for metrics
  all_metrics <- data.frame(
    beta_est = numeric(n_sim),
    beta_bias = numeric(n_sim),
    power = numeric(n_sim),
    coverage = numeric(n_sim)
  )

  for (sim in 1:n_sim) {
    # Generate random cluster-level effects
    cluster_effects <- rnorm(n_clusters, mean = 0, sd = sqrt(gamma2))
    # Generate cluster-level means
    cluster_means <- alpha + beta * cluster_treatment + cluster_effects

    # Simulate data
    cluster_data <- data.frame(
      cluster_id = rep(1:n_clusters, each = n_obs_per_cluster)[1:(n_clusters * n_obs_per_cluster)],
      X = rep(cluster_treatment, each = n_obs_per_cluster)[1:(n_clusters * n_obs_per_cluster)],
      Y = rnorm(
        n_clusters * n_obs_per_cluster,
        mean = cluster_means[rep(1:n_clusters, each = n_obs_per_cluster)],
        sd = sqrt(sigma2)
      )[1:(n_clusters * n_obs_per_cluster)]
    )

    # Fit a linear mixed-effects model
    model <- lmerTest::lmer(Y ~ X + (1 | cluster_id), data = cluster_data)

    # Extract measurements
    beta_est <- fixef(model)["X"]
```

```
    ci <- confint(model, parm = "X", method = "Wald")
    ci_coverage <- (beta >= ci[1] & beta <= ci[2])
    p_value <- summary(model)$coefficients["X", "Pr(>|t|)"]

    # Store simulation results
    all_metrics[sim, ] <- c(
      beta_est = beta_est,
      beta_bias = beta_est - beta,
      power = ifelse(p_value < alpha_level, 1, 0),
      coverage = ci_coverage
    )
  }

  # Compute performance metrics
  beta_est_var <- var(all_metrics$beta_est)
  min_beta_est <- min(all_metrics$beta_est)
  max_beta_est <- max(all_metrics$beta_est)
  avg_metrics <- colMeans(all_metrics)

  # Return results
  results <- data.frame(
    n_clusters = n_clusters,
    n_obs_per_cluster = n_obs_per_cluster,
    true_beta = beta,
    beta_est_mean = avg_metrics["beta_est"],
    min_beta_est = min_beta_est,
    max_beta_est = max_beta_est,
    beta_bias_mean = avg_metrics["beta_bias"],
    beta_est_var = beta_est_var,
    power = avg_metrics["power"],
    ci_coverage = avg_metrics["coverage"]
  )

  return(list(
    metrics = results,
    simulated_data = cluster_data
  ))
}

sim_poisson <- function(n_clusters, B, c1, c1_c2_ratio, alpha, beta, gamma2, n_sim, alpha_level = 0.05)
  set.seed(2550)
  # calculate c2 from the ratio
  c2 <- c1 / c1_c2_ratio

  # calculate the number of observations per cluster
  n_obs_per_cluster <- floor((B - n_clusters * c1) / (n_clusters * c2)) + 1


  # assign clusters to treatment (X = 1) or control (X = 0)
  cluster_treatment <- rep(c(0,1), n_clusters/2)

  # initialize data frame
  all_metrics <- data.frame(
```

```
  beta_est = numeric(n_sim),
  beta_bias = numeric(n_sim),
  power = numeric(n_sim),
  coverage = numeric(n_sim)
)

for (sim in 1:n_sim) {

  # generate random cluster-level effects (log scale)
  cluster_effects <- rnorm(n_clusters, mean = alpha, sd = sqrt(gamma2))
  # generate cluster-level means (log scale)
  log_mu <- alpha + beta * cluster_treatment + cluster_effects
  mu <- exp(log_mu)

cluster_data <- data.frame(
cluster_id = rep(1:n_clusters, each = n_obs_per_cluster)[1:(n_clusters * n_obs_per_cluster)],
X = rep(cluster_treatment, each = n_obs_per_cluster)[1:(n_clusters * n_obs_per_cluster)],
Y = rpois(n_clusters * n_obs_per_cluster, lambda = mu[rep(1:n_clusters, each = n_obs_per_cluster)])[1

  # fit a Poisson GLM with random intercept for clusters
  model <- lme4::glmer(Y ~ X + (1 | cluster_id), data = cluster_data, family = poisson())

  # extract measurements
  beta_est <- fixef(model)["X"]
  ci <- confint(model, parm = "X", method = "Wald")
  ci_coverage <- (beta >= ci[1] & beta <= ci[2])
  p_value <- summary(model)$coefficients["X", "Pr(>|z|)"]

  # store simulation results
  all_metrics[sim, ] <- c(
    beta_est = beta_est,
    beta_bias = beta_est - beta,
    power = ifelse(p_value < alpha_level, 1, 0),
    coverage = ci_coverage
  )
}

# Compute performance
beta_est_var <- var(all_metrics$beta_est)
min_beta_est <- min(all_metrics$beta_est)
max_beta_est <- max(all_metrics$beta_est)
avg_metrics <- colMeans(all_metrics)

# Return results
results <- data.frame(
  n_clusters = n_clusters,
  n_obs_per_cluster = n_obs_per_cluster,
  true_beta = beta,
  beta_est_mean = avg_metrics["beta_est"],
  min_beta_est = min_beta_est,
  max_beta_est = max_beta_est,
  beta_bias_mean = avg_metrics["beta_bias"],
  beta_est_var = beta_est_var,
```

```
    power = avg_metrics["power"],
    ci_coverage = avg_metrics["coverage"]
  )

  return(list(
    metrics = results,
    simulated_data = cluster_data
  ))
}
# Function to find optimal # of clusters and # of obs/cluster under Normal
sim_normal_opt <- function(n_clusters_seq, c1_c2_ratios, B, c1, alpha, beta, sigma2, gamma2, n_sim, alp

  # Initialize storage for all results
  all_results <- list()

  # Loop over parameters (# of clusters and cost ratio)
  for (n_clusters in n_clusters_seq) {
    for (c1_c2_ratio in c1_c2_ratios) {
      # Run simulation for current parameter set
      result <- sim_normal(
        n_clusters = n_clusters,
        B = B,
        c1 = c1,
        c1_c2_ratio = c1_c2_ratio,
        alpha = alpha,
        beta = beta,
        gamma2 = gamma2,
        sigma2 = sigma2,
        n_sim = n_sim,
        alpha_level = alpha_level
      )

      # Store the results
      all_results <- append(all_results, list(
        result$metrics %>% mutate(c1_c2_ratio = c1_c2_ratio)
      ))
    }
  }

  # Combine all results into a single data frame
  combined_results <- bind_rows(all_results)
  return(combined_results)
}

# Set Parameters
n_clusters_seq <- seq(10, 50, 5)
c1_c2_ratios <- c(2, 5, 10, 20)

B <- 2000
c1 <- 20
alpha <- 2
beta <- 1.5
sigma2 <- 1
```

```r
gamma2 <- 1
n_sim <- 100

# Run simulations over the grid
# res_normal_opt <- sim_normal_opt(
#   n_clusters_seq = n_clusters_seq,
#   c1_c2_ratios = c1_c2_ratios,
#   B = B,
#   c1 = c1,
#   alpha = alpha,
#   beta = beta,
#   sigma2 = sigma2,
#   gamma2 = gamma2,
#   n_sim = n_sim
# )
# # Store as .csv file
# write.csv(res_normal_opt, "res_normal_opt.csv")
res_normal_opt <- read.csv("res_normal_opt.csv")
# Function to find optimal # of clusters and # of obs/cluster under Poisson
sim_poisson_opt <- function(n_clusters_seq, c1_c2_ratios, B, c1, alpha, beta, gamma2, n_sim, alpha_level
  all_results <- list()

  # Loop over the parameter grid
  for (c1_c2_ratio in c1_c2_ratios) {
    for (n_clusters in n_clusters_seq) {
      # Run the simulation for each configuration
      result <- sim_poisson(
        n_clusters = n_clusters,
        B = B,
        c1 = c1,
        c1_c2_ratio = c1_c2_ratio,
        alpha = alpha,
        beta = beta,
        gamma2 = gamma2,
        n_sim = n_sim,
        alpha_level = alpha_level
      )

      # Store the results
      all_results <- append(all_results, list(
        result$metrics %>% mutate(c1_c2_ratio = c1_c2_ratio)
      ))
    }
  }

  # Combine all results into a single data frame
  combined_results <- bind_rows(all_results)
  return(combined_results)
}

# Set Parameters
n_clusters_seq <- seq(10, 50, 5)
c1_c2_ratios <- c(2, 5, 10, 20)
```

```r
# Run the evaluation
# res_poisson_opt <- sim_poisson_opt(
#   n_clusters_seq = n_clusters_seq,
#   c1_c2_ratios = c1_c2_ratios,
#   B = 2000,
#   c1 = 20,
#   alpha = 2,
#   beta = 1.5,
#   gamma2 = 1,
#   n_sim = 100,
#   alpha_level = 0.05
# )
# # Store as .csv file
# write.csv(res_poisson_opt, "res_poisson_opt.csv")
res_poisson_opt <- read.csv("res_poisson_opt.csv")
# Results Table (Normal)
r2_df <- res_normal_opt %>%
  filter(c1_c2_ratio == 2) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`G` = n_clusters, `# of Obs / Cluster (2)` = n_obs_per_cluster, `Var(beta_est) (2)` = beta_est

r5_df <- res_normal_opt %>%
  filter(c1_c2_ratio == 5) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`# of Obs / Cluster (5)` = n_obs_per_cluster, `Var(beta_est) (5)` = beta_est_var)

r10_df <- res_normal_opt %>%
  filter(c1_c2_ratio == 10) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`# of Obs / Cluster (10)` = n_obs_per_cluster, `Var(beta_est) (10)` = beta_est_var)

r20_df <- res_normal_opt %>%
  filter(c1_c2_ratio == 20) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`# of Obs / Cluster (20)` = n_obs_per_cluster, `Var(beta_est) (20)` = beta_est_var)

combined_df <- bind_cols(r2_df, r5_df, r10_df, r20_df)

# Results Table (Poisson)
r2_df_poisson <- res_poisson_opt %>%
  filter(c1_c2_ratio == 2) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`G` = n_clusters, `# of Obs / Cluster (2)` = n_obs_per_cluster, `Var(beta_est) (2)` = beta_est

r5_df_poisson <- res_poisson_opt %>%
  filter(c1_c2_ratio == 5) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`# of Obs / Cluster (5)` = n_obs_per_cluster, `Var(beta_est) (5)` = beta_est_var)

r10_df_poisson <- res_poisson_opt %>%
  filter(c1_c2_ratio == 10) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`# of Obs / Cluster (10)` = n_obs_per_cluster, `Var(beta_est) (10)` = beta_est_var)
```

```r
r20_df_poisson <- res_poisson_opt %>%
  filter(c1_c2_ratio == 20) %>%
  mutate(beta_est_var = round(beta_est_var, 2)) %>%
  select(`# of Obs / Cluster (20)` = n_obs_per_cluster, `Var(beta_est) (20)` = beta_est_var)

# Combined Table
combined_df_poisson <- bind_cols(r2_df_poisson, r5_df_poisson, r10_df_poisson, r20_df_poisson)

stacked_combined_df <- bind_rows(combined_df, combined_df_poisson)

stacked_combined_df %>%
  gt() %>%
  tab_row_group(
    group = "Poisson",
    rows = 10:18
  ) %>%
    tab_row_group(
    group = "Normal",
    rows = 1:9
  ) %>%
  tab_spanner(
    label = "c1/c2=2",
    columns = c(`# of Obs / Cluster (2)`, `Var(beta_est) (2)`)
  ) %>%
  tab_spanner(
    label = "c1/c2=5",
    columns = c(`# of Obs / Cluster (5)`, `Var(beta_est) (5)`)
  ) %>%
  tab_spanner(
    label = "c1/c2=10",
    columns = c(`# of Obs / Cluster (10)`, `Var(beta_est) (10)`)
  ) %>%
  tab_spanner(
    label = "c1/c2=20",
    columns = c(`# of Obs / Cluster (20)`, `Var(beta_est) (20)`)
  ) %>%
  cols_label(
    `# of Obs / Cluster (2)` = "R",
    `# of Obs / Cluster (5)` = "R",
    `# of Obs / Cluster (10)` = "R",
    `# of Obs / Cluster (20)` = "R",
    `Var(beta_est) (2)` = "Var(beta_hat)",
    `Var(beta_est) (5)` = "Var(beta_hat)",
    `Var(beta_est) (10)` = "Var(beta_hat)",
    `Var(beta_est) (20)` = "Var(beta_hat)"
  ) %>%
  tab_options(
    table.font.size = px(8),
    heading.title.font.size = px(8)
  ) %>%
  tab_style(
    style = cell_text(weight = "bold", align = "center"),
    locations = cells_column_labels(everything())
```

```
  ) %>%
  tab_style(
    style = list(
      cell_fill(color = "#c9ecb4")),
    locations = cells_body(columns = c("G","# of Obs / Cluster (2)", "Var(beta_est) (2)",

                                       "# of Obs / Cluster (20)", "Var(beta_est) (20)"), rows = 9)
  ) %>%
  tab_style(
    style = list(
      cell_fill(color = "#c9ecb4")),
    locations = cells_body(columns = c("G","# of Obs / Cluster (10)", "Var(beta_est) (10)",
                                       "# of Obs / Cluster (5)", "Var(beta_est) (5)"), rows = 8)
  ) %>%
  tab_style(
    style = list(
      cell_fill(color = "#c9ecb4")),
    locations = cells_body(columns = c("G",
                                       "# of Obs / Cluster (5)", "Var(beta_est) (5)",
                                       "# of Obs / Cluster (20)", "Var(beta_est) (20)"), rows =18)) %>%
  tab_style(
    style = list(
      cell_fill(color = "#c9ecb4")),
    locations = cells_body(columns = c("G","# of Obs / Cluster (2)", "Var(beta_est) (2)",
                                       "# of Obs / Cluster (10)", "Var(beta_est) (10)"), rows = 17)
  ) %>%
  tab_style(
    style = cell_text(weight = "bold", align = "center"), # Bold the spanner text
    locations = cells_column_spanners(everything())
  ) %>%
  tab_style(
    style = cell_text(weight = "bold"), # Bold the row group labels
    locations = cells_row_groups()
  ) %>%
  tab_header(
    title = "Table 1: Simulation Results Vary by Number of Clusters and Relative Costs"
  )
# Assign colors for the c1/c2 ratio
colors_normal <- c(
  "2" = brewer.pal(9, "YlOrRd")[4],
  "5" = brewer.pal(9, "YlOrRd")[6],
  "10" = brewer.pal(9, "YlOrRd")[8],
  "20" = brewer.pal(9, "YlOrRd")[9]
)


# Plot with adjusted text position
ggplot(res_normal_opt, aes(x = n_clusters, y = beta_est_var)) +
  geom_line(aes(color = as.factor(c1_c2_ratio)), size = 0.8) +
  geom_point(aes(color = as.factor(c1_c2_ratio)), size = 1) +
  geom_text(aes(label = n_obs_per_cluster, color = as.factor(c1_c2_ratio)),
            position = position_nudge(y = 0.025, x=2.2),  # Nudging text upward
            size = 2, check_overlap = TRUE) +
```

```
  scale_color_manual(values = colors_normal, name = "Relative costs (c1/c2)") +
  facet_wrap(~c1_c2_ratio, nrow = 1) +
  labs(
    title = "Figure 1: Variance of Beta Estimates vs. Number of Clusters",
    x = "Number of Clusters",
    y = "Variance of Beta Estimates"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8),
    legend.position = "bottom"
  )
# Assign colors for the c1/c2 ratio
colors_poisson <- c(
  "2" = brewer.pal(9, "Blues")[5],  # Moderate orange
  "5" = brewer.pal(9, "Blues")[7],  # Deeper orange-red
  "10" = brewer.pal(9, "Blues")[8],  # Dark red
  "20" = brewer.pal(9, "Blues")[9]
)


# Plot with adjusted text position
ggplot(res_poisson_opt, aes(x = n_clusters, y = beta_est_var)) +
  geom_line(aes(color = as.factor(c1_c2_ratio)), size = 0.8) +
  geom_point(aes(color = as.factor(c1_c2_ratio)), size = 1) +
  geom_text(aes(label = n_obs_per_cluster, color = as.factor(c1_c2_ratio)),
            position = position_nudge(y = 0.025, x=2.2),  # Nudging text upward
            size = 2, check_overlap = TRUE) +
  scale_color_manual(values = colors_poisson, name = "Relative costs (c1/c2)") +
  facet_wrap(~c1_c2_ratio, nrow = 1) +
  labs(
    title = "Figure 2: Variance of Beta Estimates vs. Number of Clusters",
    x = "Number of Clusters",
    y = "Variance of Beta Estimates"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
    axis.title = element_text(size = 10),
    axis.text = element_text(size = 8),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8),
    legend.position = "bottom"
  )
# Function to simulate for different sigma2 and beta under optimal # of clusters and # of obs/cluster
sim_normal_vary <- function(normal_opt, sigma2_values, beta_values, B, c1, alpha, gamma2, n_sim, alpha_

  # initialize storage for all results
  all_results <- list()
```

```r
  # Loop over within-cluster variacen and treatment effect
  for (row in seq_len(nrow(normal_opt))) {
    n_clusters <- normal_opt$n_clusters[row]
    n_obs_per_cluster <- normal_opt$n_obs_per_cluster[row]
    c1_c2_ratio <- normal_opt$c1_c2_ratio[row]

    for (sigma2 in sigma2_values) {
      for (beta in beta_values) {
        # Run simulation for current parameter set
        result <- sim_normal(
          n_clusters = n_clusters,
          B = B,
          c1 = c1,
          c1_c2_ratio = c1_c2_ratio,
          alpha = alpha,
          beta = beta,
          gamma2 = gamma2,
          sigma2 = sigma2,
          n_sim = n_sim,
          alpha_level = alpha_level
        )

        all_results <- append(all_results, list(
          result$metrics %>% mutate(sigma2 = sigma2, beta = beta, c1_c2_ratio = c1_c2_ratio)
        ))
      }
    }
  }

  # Combine all results into a single data frame
  combined_results <- bind_rows(all_results)
  return(combined_results)
}

# extract optimal combination of number of cluster and number of obs / cluster
normal_opt <- res_normal_opt %>%
  group_by(c1_c2_ratio) %>%
  summarize(
    n_clusters = n_clusters[which.min(beta_est_var)],  # Optimal number of clusters
    n_obs_per_cluster = n_obs_per_cluster[which.min(beta_est_var)]
  )

# Set parameters
sigma2_values <- seq(0.1, 10, length.out=10)
beta_values <- c(0.05, 0.5, 1.5)
B <- 2000
c1 <- 20
alpha <- 2
gamma2 <- 1
n_sim <- 100

# Run simulations
# res_normal_vary <- sim_normal_vary(
```

```r
#   normal_opt = normal_opt,
#   sigma2_values = sigma2_values,
#   beta_values = beta_values,
#   B = B,
#   c1 = c1,
#   alpha = alpha,
#   gamma2 = gamma2,
#   n_sim = n_sim
# )
#
# write.csv(res_normal_vary, "res_normal_vary.csv")
res_normal_vary <- read.csv("res_normal_vary.csv")
# Function to simulate for different sigma2 and beta under optimal # of clusters and # of obs/cluster
sim_poisson_vary <- function(poisson_opt, gamma2_values, beta_values, B, c1, alpha, n_sim, alpha_level)
  all_results <- list()

  # Loop over between-cluster variance and treatment effect size
  for (row in seq_len(nrow(poisson_opt))) {
    n_clusters <- poisson_opt$n_clusters[row]
    n_obs_per_cluster <- poisson_opt$n_obs_per_cluster[row]
    c1_c2_ratio <- poisson_opt$c1_c2_ratio[row]

    for (gamma2 in gamma2_values) {
      for (beta in beta_values) {
        result <- sim_poisson(
          n_clusters = n_clusters,
          B = B,
          c1 = c1,
          c1_c2_ratio = c1_c2_ratio,
          alpha = alpha,
          beta = beta,
          gamma2 = gamma2,
          n_sim = n_sim,
          alpha_level = alpha_level
        )

        all_results <- append(all_results, list(
          result$metrics %>% mutate(gamma2 = gamma2, beta = beta, c1_c2_ratio = c1_c2_ratio)
        ))
      }
    }
  }

  combined_results <- bind_rows(all_results)
  return(combined_results)
}

# Set parameters
 beta_values <- c(0.05, 0.5, 1.5)
# beta_values <- c(0.5, 1.5, 3)
gamma2_values <- seq(0.1, 10, length.out = 10)

# Extract optimal configurations from varying_results
```

```r
poisson_opt <- res_poisson_opt %>%
  group_by(c1_c2_ratio) %>%
  summarize(
    n_clusters = n_clusters[which.min(beta_est_var)],  # Optimal number of clusters
    n_obs_per_cluster = n_obs_per_cluster[which.min(beta_est_var)]
  )

# Run the simulation
# res_poisson_vary <- sim_poisson_vary(
#   poisson_opt = poisson_opt,
#   gamma2_values = gamma2_values,
#   beta_values = beta_values,
#   B = 2000,
#   c1 = 20,
#   alpha = alpha,
#   n_sim = 100,
#   alpha_level = 0.05
# )
#
# write.csv(res_poisson_vary, "res_poisson_vary.csv")
res_poisson_vary <- read.csv("res_poisson_vary.csv")
# Results Table (Normal)
df_normal_vary <- res_normal_vary %>%
  arrange(c1_c2_ratio, sigma2, beta) %>%
  filter( sigma2 %in% c(0.1, 4.5, 10)) %>%
  mutate(beta_est_var = round(beta_est_var, 2),
         `G (R) (n)`=paste(`n_clusters`, "(", `n_obs_per_cluster`, ")")) %>%
  select(c1_c2_ratio, `sigma^2 (gamma^2)`= sigma2, beta, `G (R) (n)`, `Var(beta_est) (n)` = beta_est_va

# Results Table (Poisson)
df_poisson_vary <- res_poisson_vary %>%
  arrange(c1_c2_ratio, gamma2, beta) %>%
  filter(gamma2 %in% c(0.1, 4.5, 10)) %>%
  mutate(beta_est_var = round(beta_est_var, 2),
         `G (R) (p)`=paste(`n_clusters`, "(", `n_obs_per_cluster`, ")")) %>%
  select(`G (R) (p)`,`Var(beta_est) (p)` = beta_est_var, `power (p)` = power)

df_com_vary <- bind_cols(df_normal_vary, df_poisson_vary)
df_com_vary %>%
  gt() %>%
  tab_spanner(
    label = "Normal",
    columns = c(`G (R) (n)`, `Var(beta_est) (n)`, `power (n)`)
  ) %>%
  tab_spanner(
    label = "Poisson",
    columns = c(`G (R) (p)`, `Var(beta_est) (p)`, `power (p)`)
  ) %>%
  cols_label(
    c1_c2_ratio = "c1/c2",
    #`sigma^2 (gamma^2)` = md("&sigma;<sup>2</sup> (&gamma;<sup>2</sup>)"),
    #beta = md("&beta;"),
    `G (R) (n)` = "G (R)",
```

```
    `G (R) (p)` = "G (R)",
    # `Var(beta_est) (n)` = md("Var(&beta;<sub>est</sub>)"),
    # `Var(beta_est) (p)` = md("Var(&beta;<sub>est</sub>)"),
    `Var(beta_est) (n)` = "Var(beta_hat)",
    `Var(beta_est) (p)` = "Var(beta_hat)",
    `power (n)` = "power",
    `power (p)` = "power"
  ) %>%
  tab_options(
    table.font.size = px(8),
    heading.title.font.size = px(8)
  ) %>%
  tab_style(
    style = cell_text(weight = "bold", align = "center"),
    locations = cells_column_labels(everything())
  ) %>%
  tab_style(
    style = cell_text(weight = "bold", align = "center"), # Bold the spanner text
    locations = cells_column_spanners(everything())
  ) %>%
  tab_header(
    title = "Table 2: Simulation Results Vary by Variance and Treatment Effect"
  )
final_results_long_normal <- res_normal_vary %>%
  pivot_longer(cols = c(beta_est_var, power, ci_coverage),
               names_to = "metric",
               values_to = "value")

# Define y-axis limits for each metric
y_lim_normal <- list(
  power = c(0, 1),
  beta_est_var = c(0, max(res_normal_vary$beta_est_var, na.rm = TRUE)),
  avg_ci_coverage = c(0.8, 1)
)

colors_normal <- c(
  "2" = brewer.pal(9, "YlOrRd")[5],  # Moderate orange
  "5" = brewer.pal(9, "YlOrRd")[7],  # Deeper orange-red
  "10" = brewer.pal(9, "YlOrRd")[8],  # Dark red
  "20" = brewer.pal(9, "YlOrRd")[9]
)

# Updated function to create a plot for a given metric
plot_metric_consistent_normal <- function(metric, metric_label) {
  ggplot(final_results_long_normal %>% filter(metric == !!metric),
         aes(x = sigma2, y = value, color = as.factor(c1_c2_ratio))) +
    geom_line(size = 0.3, alpha=0.8) +
    geom_point(size = 0.6) +
    facet_wrap(~ beta, scales = "fixed") +
    scale_color_manual(values = colors_normal,
                       name = "c1/c2 Ratio") +
    coord_cartesian(ylim = y_lim_normal[[metric]]) +
    labs(
```

```
      #title = paste(metric_label, "by Gamma^2, Colored by c1/c2 Ratio"),
      x = expression(sigma^2),
      y = metric_label
    ) +
    theme_minimal() +
    theme(
      axis.text = element_text(size = 6),
      axis.title = element_text(size = 8, face = "bold"),
      plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
      legend.position = "bottom"
    )
}


# Generate plots for each metric
plot_beta_est_var_normal <- plot_metric_consistent_normal("beta_est_var", "Variance of Beta Estimate")
plot_power_normal <- plot_metric_consistent_normal("power", "Power")

# Display the plots (one at a time)
combined_plot_normal <- ggarrange(plot_beta_est_var_normal, plot_power_normal, common.legend = TRUE, #
  legend = "bottom")

final_plot_normal <- annotate_figure(
  combined_plot_normal,
  top = text_grob("Figure 3: Variance of Beta Estimate and Power Across c1/c2 Ratios", size = 10, face =
)
final_plot_normal
# Step 4: Visualize results
final_results_long_poisson <- res_poisson_vary %>%
  pivot_longer(cols = c(beta_est_var, power, ci_coverage),
               names_to = "metric",
               values_to = "value")

# Define consistent y-limits for each metric
y_lim_poisson <- list(
  beta_est_var = c(0, max(res_poisson_vary$beta_est_var, na.rm = TRUE)),
  power = c(0, 1),
  ci_coverage = c(0.8, 1)
)

colors_poisson <- c(
  "2" = brewer.pal(9, "Blues")[5],  # Moderate orange
  "5" = brewer.pal(9, "Blues")[7],  # Deeper orange-red
  "10" = brewer.pal(9, "Blues")[8],  # Dark red
  "20" = brewer.pal(9, "Blues")[9]
)

plot_metric_consistent_y <- function(metric, metric_label) {
  ggplot(final_results_long_poisson %>% filter(metric == !!metric),
         aes(x = gamma2, y = value, color = as.factor(c1_c2_ratio))) +
    geom_line(size = 0.3, alpha=0.8) +
    geom_point(size = 0.6) +
    facet_wrap(~ beta, scales = "fixed") +
```

```
    scale_color_manual(values = colors_poisson,
                       name = "c1/c2 Ratio") +
    coord_cartesian(ylim = y_lim_poisson[[metric]]) +
    labs(
      #title = paste(metric_label, "by Gamma^2, Colored by c1/c2 Ratio"),
      x = expression(gamma^2),
      y = metric_label
    ) +
    theme_minimal() +
    theme(
      axis.text = element_text(size = 6),
      axis.title = element_text(size = 8, face = "bold"),
      plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
      legend.position = "bottom"
    )
}

# Plot beta_est_var
plot_beta_est_var_poisson <- plot_metric_consistent_y("beta_est_var", "Variance of Beta Estimate")
plot_power_poisson <- plot_metric_consistent_y("power", "Power")

# Display the plots (one at a time)
combined_plot_poisson <- ggarrange(plot_beta_est_var_poisson, plot_power_poisson, common.legend = TRUE,
  legend = "bottom")

final_plot_poisson <- annotate_figure(
  combined_plot_poisson,
  top = text_grob("Figure 4: Variance of Beta Estimate and Power Across c1/c2 Ratios", size = 10, face =
)
final_plot_poisson
```