CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

Cluster Sampling – Population Mean Estimation

Subject:

Sampling Techniques

Submitted By:

Ann Maria Anil

2341419

6BScDS

Submitted To:

Prof.Dibu

Date of Submission:

16th December, 2025

1. Introduction

Cluster sampling is a probability-based survey technique where the population is divided into mutually exclusive groups called clusters. Instead of sampling individual units from the entire population, a subset of clusters is randomly selected, and all units within these clusters are included in the survey. This approach is especially useful when the population is large, geographically dispersed, or when surveying every unit individually is costly or time-consuming.

This report demonstrates a Shiny-based application for designing a cluster sampling survey, estimating population mean, and visualizing the selected clusters.

2. Objective

The main objectives of this cluster sampling exercise are:

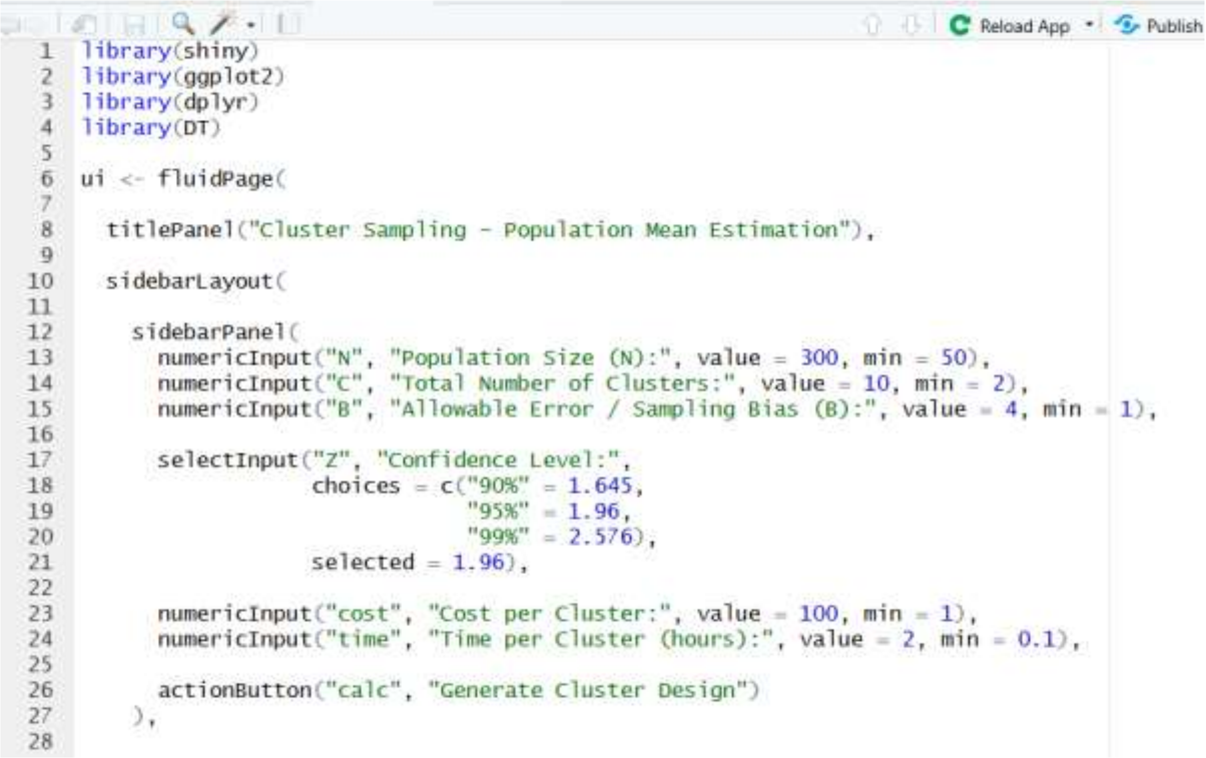
- To determine the **number of clusters** to include in the survey based on user-defined parameters such as allowable error and confidence level.
- To calculate **expected cost and time** for surveying the selected clusters.
- To provide a **visual representation** of the survey design for better understanding and planning.
- To create an **interactive tool** that allows users to experiment with different survey parameters and immediately see the effect on cluster selection.

3. Methodology

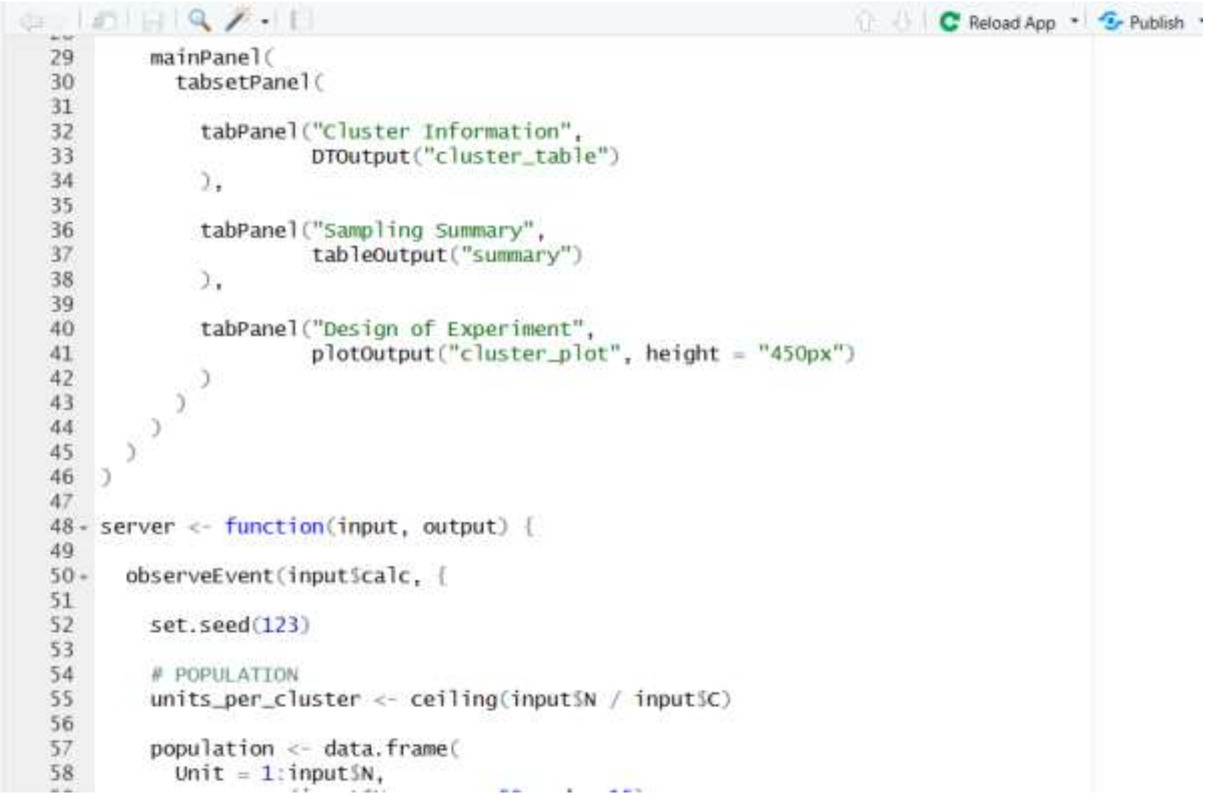
1. **Population and Clusters Setup:**
    - The total population is divided into a chosen number of clusters.
    - Each cluster contains a number of units, and each unit has an associated observed value (simulated in the app for demonstration purposes).
    - Cost and time per cluster are defined to allow estimation of survey resources.
2. **Cluster Summary:**
    - For each cluster, the app computes:
        - Cluster size (number of units)
        - Mean of observed values
        - Variance of observed values
        - Cost and time

3. **Cluster Selection:**
   - Based on the allowable error (sampling bias) and confidence level, the app determines how many clusters should be selected.
   - Clusters are selected randomly to ensure unbiased sampling.
4. **Cost & Time Calculation:**
   - Total expected cost and survey time are calculated by summing the cost and time across the selected clusters.
5. **Visualization:**
   - The design of experiment is plotted using a scatter plot faceted by cluster.
   - Units within selected clusters are highlighted in a distinctive color, while non-selected clusters are shown in a neutral color.
   - This visual representation helps quickly understand which clusters are included and the overall survey coverage.

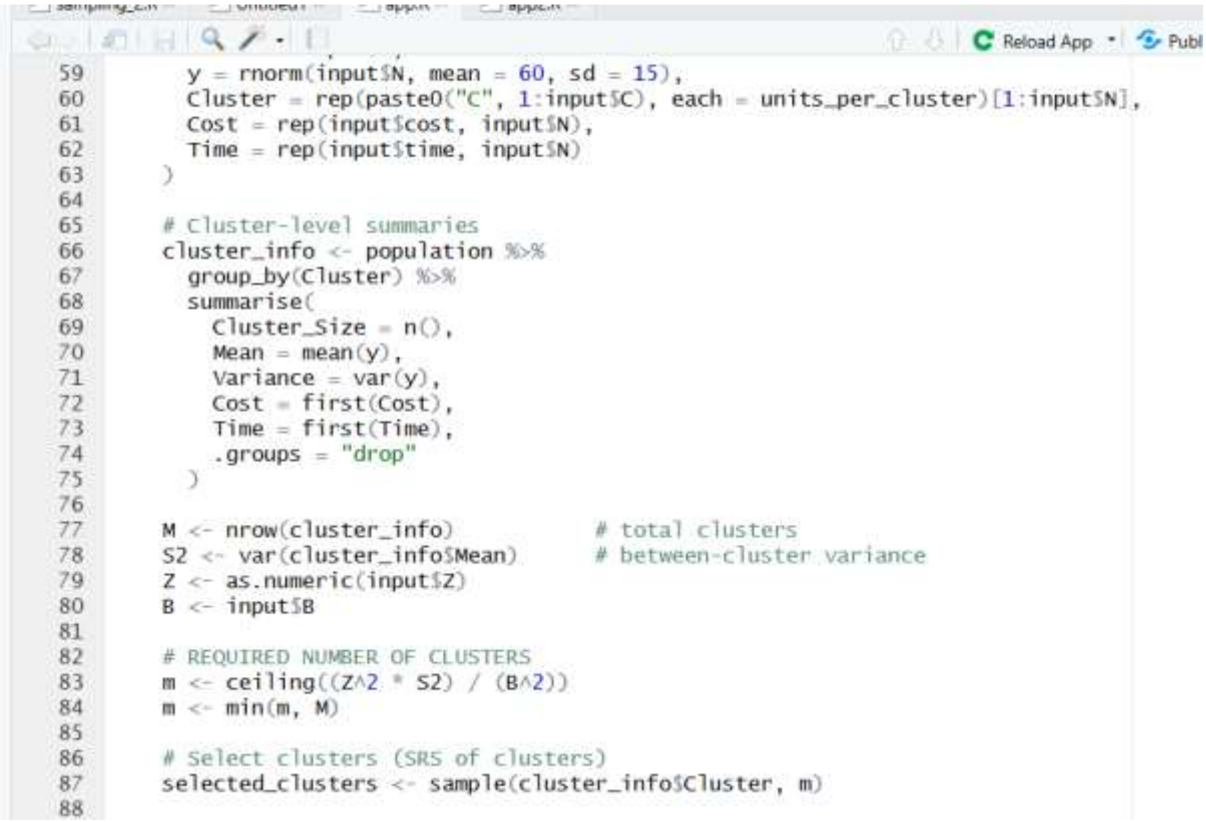
4. Features of the Shiny App



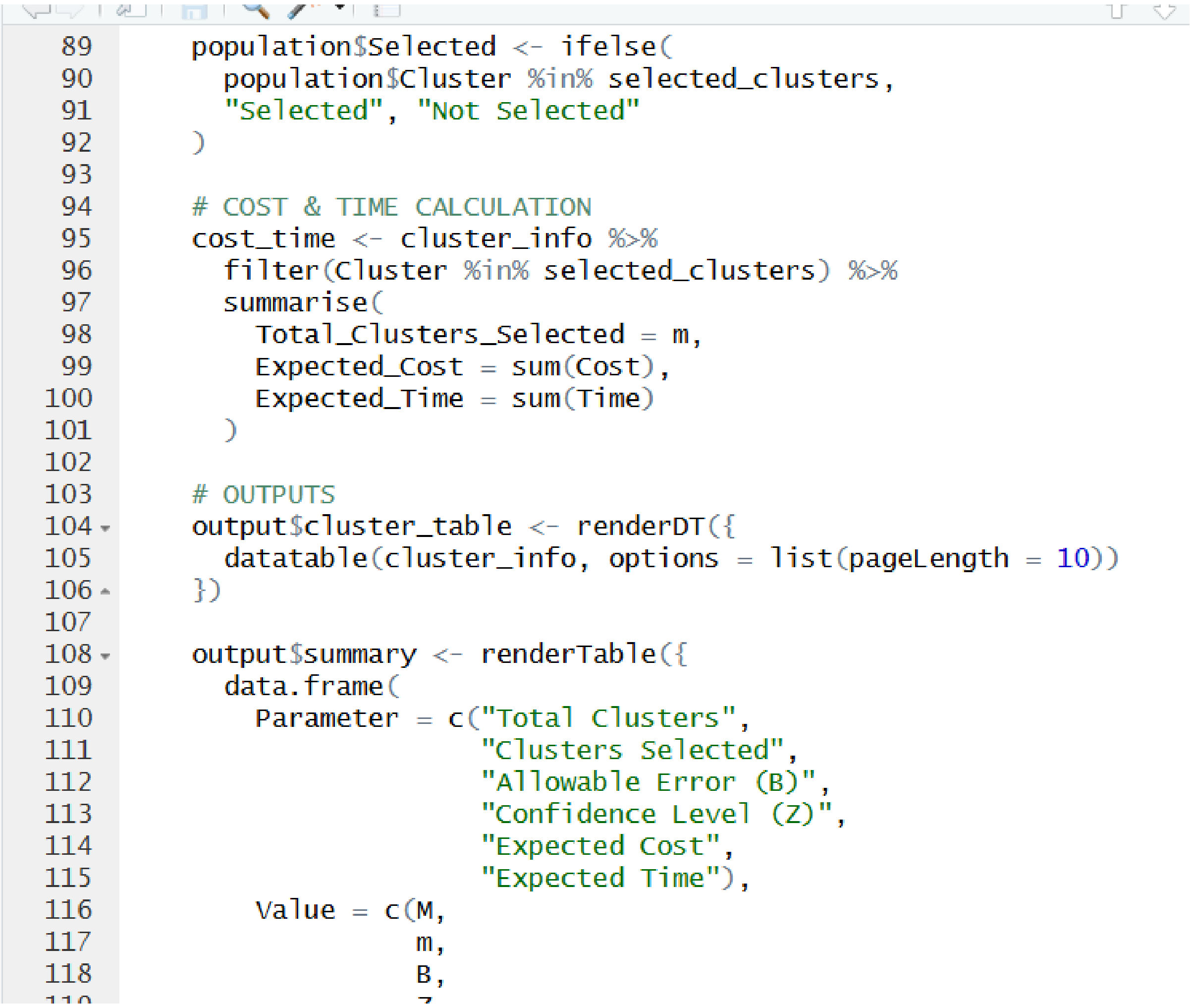
```r
library(shiny)
library(ggplot2)
library(dplyr)
library(DT)

ui <- fluidPage(

  titlePanel("Cluster Sampling - Population Mean Estimation"),

  sidebarLayout(

    sidebarPanel(
      numericInput("N", "Population Size (N):", value = 300, min = 50),
      numericInput("C", "Total Number of Clusters:", value = 10, min = 2),
      numericInput("B", "Allowable Error / Sampling Bias (B):", value = 4, min = 1),

      selectInput("Z", "Confidence Level:",
                  choices = c("90%" = 1.645,
                              "95%" = 1.96,
                              "99%" = 2.576),
                  selected = 1.96),

      numericInput("cost", "Cost per Cluster:", value = 100, min = 1),
      numericInput("time", "Time per Cluster (hours):", value = 2, min = 0.1),

      actionButton("calc", "Generate Cluster Design")
    ),

```



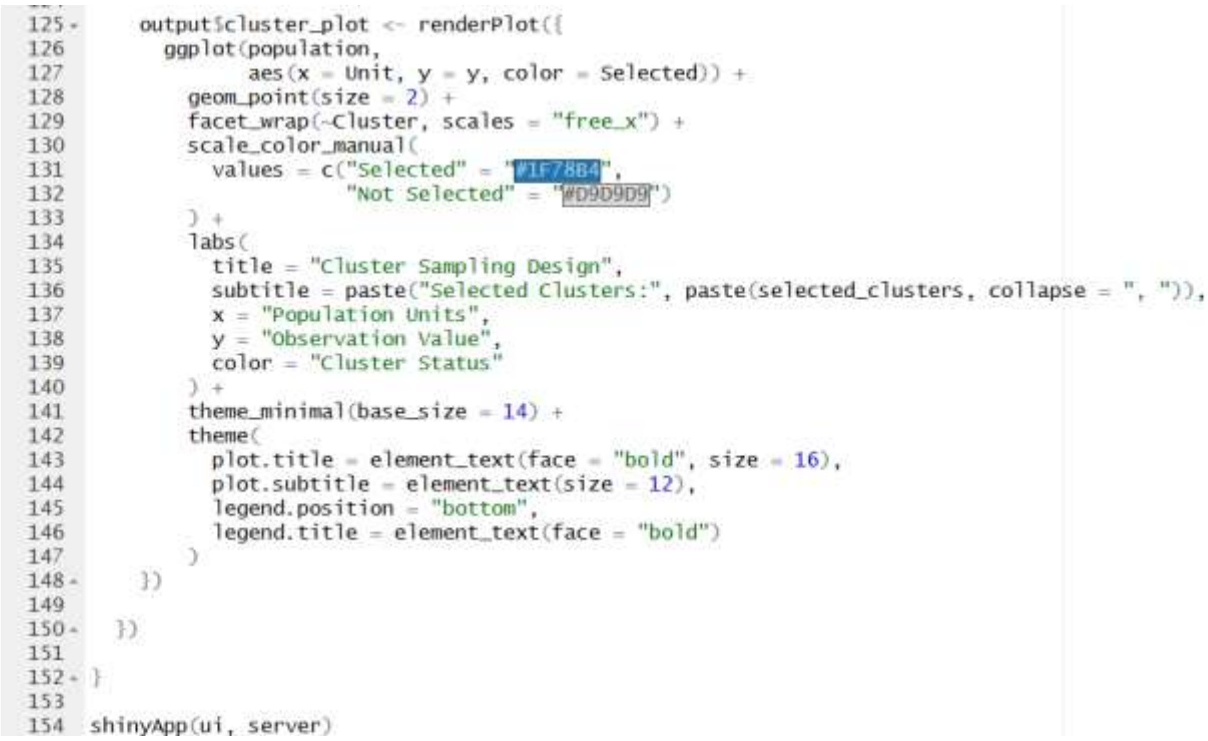
```r
    mainPanel(
      tabsetPanel(

        tabPanel("Cluster Information",
                 DTOutput("cluster_table")
        ),

        tabPanel("Sampling Summary",
                 tableOutput("summary")
        ),

        tabPanel("Design of Experiment",
                 plotOutput("cluster_plot", height = "450px")
        )
      )
    )
  )
)

server <- function(input, output) {

  observeEvent(input$calc, {

    set.seed(123)

    # POPULATION
    units_per_cluster <- ceiling(input$N / input$C)

    population <- data.frame(
      Unit = 1:input$N,
```



```r
      y = rnorm(input$N, mean = 60, sd = 15),
      Cluster = rep(paste0("C", 1:input$C), each = units_per_cluster)[1:input$N],
      Cost = rep(input$cost, input$N),
      Time = rep(input$time, input$N)
    )

    # Cluster-level summaries
    cluster_info <- population %>%
      group_by(Cluster) %>%
      summarise(
        Cluster_Size = n(),
        Mean = mean(y),
        Variance = var(y),
        Cost = first(Cost),
        Time = first(Time),
        .groups = "drop"
      )

    M <- nrow(cluster_info)           # total clusters
    S2 <- var(cluster_info$Mean)      # between-cluster variance
    Z <- as.numeric(input$Z)
    B <- input$B

    # REQUIRED NUMBER OF CLUSTERS
    m <- ceiling((Z^2 * S2) / (B^2))
    m <- min(m, M)

    # Select clusters (SRS of clusters)
    selected_clusters <- sample(cluster_info$Cluster, m)

```



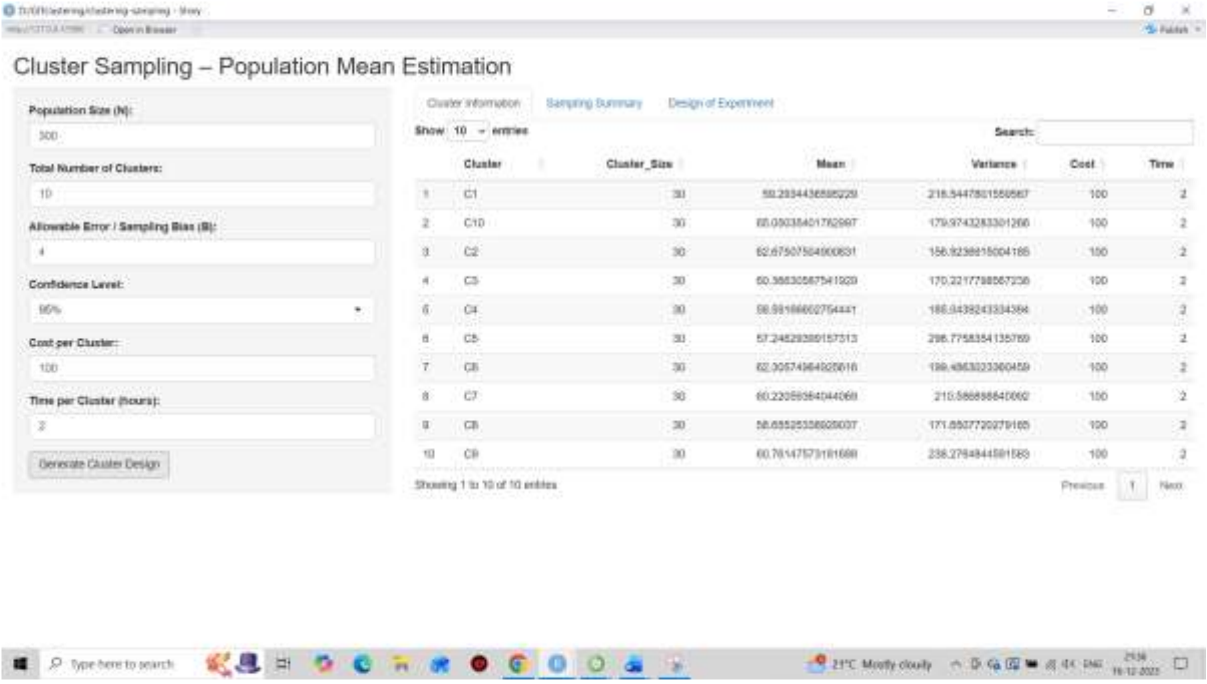
```r
    population$Selected <- ifelse(
      population$Cluster %in% selected_clusters,
      "Selected", "Not Selected"
    )

    # COST & TIME CALCULATION
    cost_time <- cluster_info %>%
      filter(Cluster %in% selected_clusters) %>%
      summarise(
        Total_Clusters_Selected = m,
        Expected_Cost = sum(Cost),
        Expected_Time = sum(Time)
      )

    # OUTPUTS
    output$cluster_table <- renderDT({
      datatable(cluster_info, options = list(pageLength = 10))
    })

    output$summary <- renderTable({
      data.frame(
        Parameter = c("Total Clusters",
                      "Clusters Selected",
                      "Allowable Error (B)",
                      "Confidence Level (Z)",
                      "Expected Cost",
                      "Expected Time"),
        Value = c(M,
                  m,
                  B,
```



```r
    output$cluster_plot <- renderPlot({
      ggplot(population,
             aes(x = Unit, y = y, color = Selected)) +
        geom_point(size = 2) +
        facet_wrap(~Cluster, scales = "free_x") +
        scale_color_manual(
          values = c("Selected" = "#1F78B4",
                     "Not Selected" = "#D9D9D9")
        ) +
        labs(
          title = "Cluster Sampling Design",
          subtitle = paste("Selected Clusters:", paste(selected_clusters, collapse = ", ")),
          x = "Population Units",
          y = "Observation Value",
          color = "Cluster Status"
        ) +
        theme_minimal(base_size = 14) +
        theme(
          plot.title = element_text(face = "bold", size = 16),
          plot.subtitle = element_text(size = 12),
          legend.position = "bottom",
          legend.title = element_text(face = "bold")
        )
    })

  })
}

shinyApp(ui, server)
```

- **Interactive Inputs:** Allows users to define population size, number of clusters, allowable error, confidence level, cost, and time per cluster.
- **Cluster Information Tab:** Displays detailed summary for each cluster in an interactive table.
- **Sampling Summary Tab:** Provides overall information including total clusters, clusters selected, allowable error, confidence level, expected cost, and time.
- **Design of Experiment Plot:** Shows a clear visualization of selected vs. non-selected clusters, facilitating planning and communication.
- **Reactivity:** Changing any input updates all outputs immediately, allowing users to experiment and observe effects dynamically.

5.Output screenshots:



Cluster Sampling – Population Mean Estimation

Population Size (N): 300

Total Number of Clusters: 10

Allowable Error / Sampling Bias (B): 4

Confidence Level: 95%

Cost per Cluster: 100

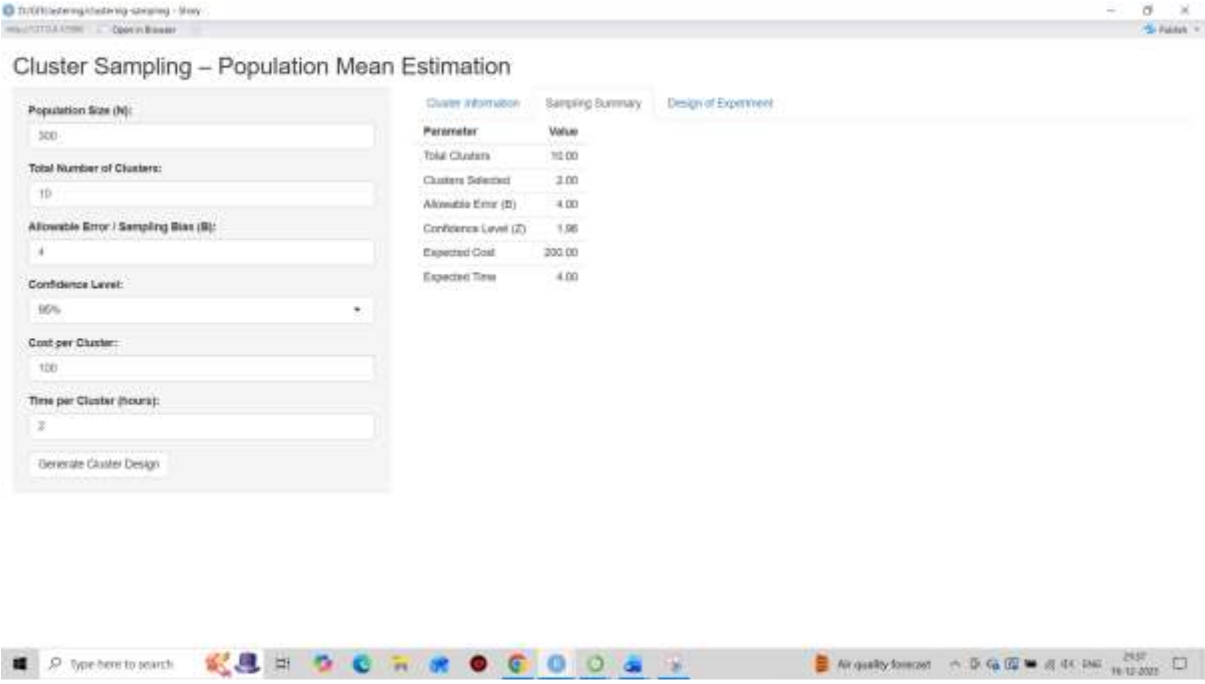
Time per Cluster (hours): 2

Show 10 entries Search:

	Cluster	Cluster_Size	Mean	Variance	Cost	Time
1	C1	30	59.2934436585229	218.5447801559567	100	2
2	C10	30	65.05038401762997	179.9743283301266	100	2
3	C2	30	62.67507504900831	156.9238615004185	100	2
4	C3	30	60.36630587541929	170.2217798567236	100	2
5	C4	30	58.99166602754441	188.9439243334384	100	2
6	C5	30	57.24829389157313	298.7758354135789	100	2
7	C6	30	62.30574964925616	199.4863023360459	100	2
8	C7	30	60.22059364044068	210.586898640062	100	2
9	C8	30	58.65525338925037	171.6507720279165	100	2
10	C9	30	60.76147573181688	238.2764844581583	100	2

Showing 1 to 10 of 10 entries

Previous 1 Next



Cluster Sampling – Population Mean Estimation

Population Size (N): 300

Total Number of Clusters: 10

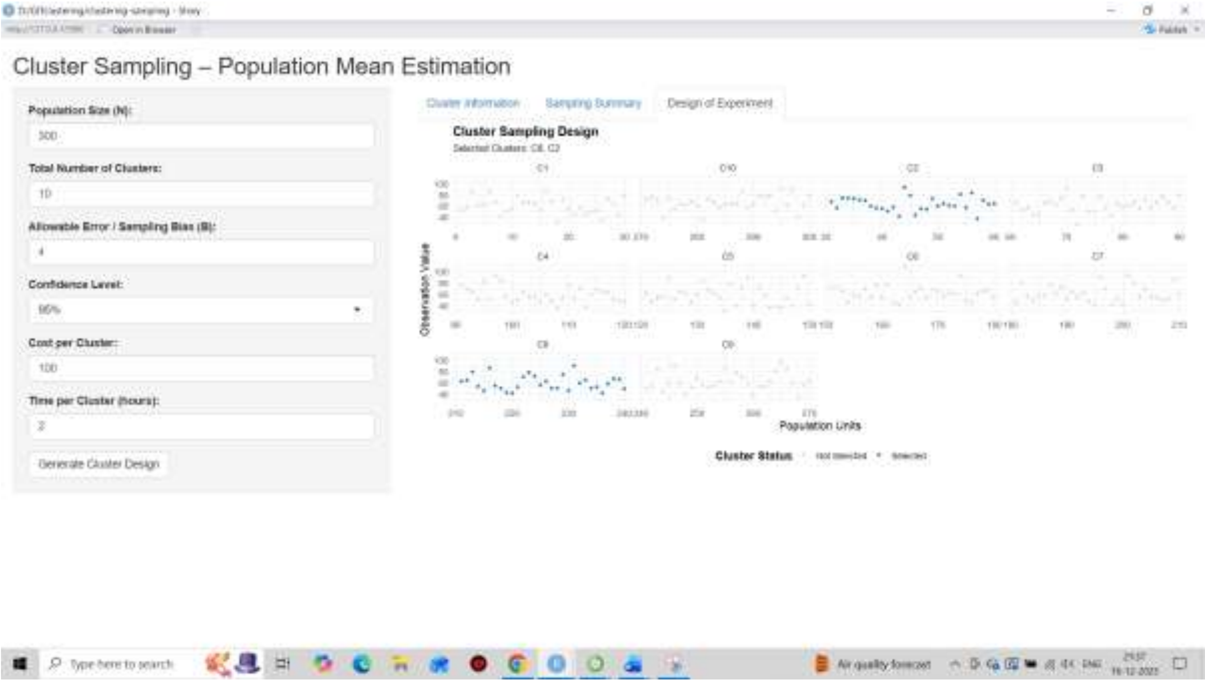
Allowable Error / Sampling Bias (B): 4

Confidence Level: 95%

Cost per Cluster: 100

Time per Cluster (hours): 2

Parameter	Value
Total Clusters	10.00
Clusters Selected	2.00
Allowable Error (B)	4.00
Confidence Level (Z)	1.96
Expected Cost	200.00
Expected Time	4.00



Cluster Sampling – Population Mean Estimation

Population Size (N): 300

Total Number of Clusters: 10

Allowable Error / Sampling Bias (B): 4

Confidence Level: 95%

Cost per Cluster: 100

Time per Cluster (hours): 2

6. Conclusion

- The Shiny app provides an **efficient and interactive tool** for designing cluster-based surveys.
- It allows users to **estimate required clusters**, **calculate expected resources**, and **visualize survey design** quickly.
- The approach ensures **efficient survey planning**, minimizing cost and time while maintaining statistical reliability.
- The visualization component makes it easier for survey designers and stakeholders to interpret and communicate the survey plan.

7.Professional github link

Including the git repository link : https://github.com/ann-maria-anil/clustering-sampling-r-shiny