Expt. No. 8

Expt. Name

Page No. 79

EXPERIMENT – 8

Aim :- Implement various sampling techniques in R and assess their impact on the accuracy of statistical parameter estimation.

Introduction :- The aim of this experiment is to implement various sampling techniques in R and asess their impact on the accuracy of statistical parameter estimation. The goal is to understand how different sampling methods influence the precision & reliability of estimated performance parameters in statistical analysis.

Software Required :-

1. R Statistical Software
2. R Studio

Relevance of Experiment :- Simpling is a fundamental step in statistical analysis, and the choice of sampling technique can significantly affect the accuracy of parameter estimation. This exp. is relevant for individuals involved in data analysis, research, and decision-making, as it provides insights into the impact of sampling methods influences the accuracy and precision of parameter estimates.

Flowchart / Pseudocode / Algorithm :-

1. Select the dataset for analysis.

Teacher's Signature:

Aim :- Implement various sampling techniques in R and assess their impact on the accuracy of statistical parameter estimation.

```r
# Load required libraries
library(dplyr)

# Load the iris dataset
data("iris")
head(iris)

# Function to calculate mean of a vector
calculate_mean <- function(vector) {
  return(mean(vector))
}

# Function to perform sampling and estimate mean
perform_sampling <- function(dataset, sample_size, sampling_method) {
  if (sampling_method == "simple_random") {
    sampled_data <- dataset[sample(nrow(dataset), sample_size, replace = FALSE), ]
  } else if (sampling_method == "stratified") {
    sampled_data <- dataset %>%
      group_by(Species) %>%
      sample_n(sample_size, replace = FALSE)
  } else if (sampling_method == "systematic") {
    # Selecting every nth observation
    n <- nrow(dataset)
    k <- sample_size
    start <- sample(1:k, 1)
    indices <- seq(start, n, by = k)
    sampled_data <- dataset[indices, ]
  } else {
    stop("Invalid sampling method.")
  }

  return(calculate_mean(sampled_data$Sepal.Length))
}

# Function to assess impact on accuracy of mean estimation
assess_accuracy <- function(dataset, sample_sizes, sampling_method, num_iterations) {
  results <- data.frame(sample_size = numeric(), mean_estimate = numeric())

  for (sample_size in sample_sizes) {
    mean_estimates <- replicate(num_iterations, perform_sampling(dataset, sample_size, sampling_method))
    mean_estimate <- mean(mean_estimates)
    result <- data.frame(sample_size = sample_size, mean_estimate = mean_estimate)
    results <- rbind(results, result)
  }

  return(results)
}

# Define parameters
sample_sizes <- c(10, 20, 30, 40, 50)
num_iterations <- 100
sampling_methods <- c("simple_random", "stratified", "systematic")

# Assess accuracy for each sampling method
accuracy_results <- lapply(sampling_methods, function(method) {
  assess_accuracy(iris, sample_sizes, method, num_iterations)
})

# Plotting the results
library(ggplot2)
library(tidyr)

accuracy_df <- bind_rows(accuracy_results, .id = "method")
ggplot(accuracy_df, aes(x = sample_size, y = mean_estimate, color = method)) +
  geom_line() +
  geom_point() +
  labs(title = "Impact of Sampling Techniques on Mean Estimation Accuracy",
       x = "Sample Size", y = "Mean Estimate") +
  theme_minimal()
```

Date

Expt. No.

Expt. Name

Page No. 20

2. Choose the sampling techniques to implement.
3. Implement the selected sampling techniques in R.
4. Estimate the desired statistical parameters for each sample.
5. Compare the parameter estimates across different sampling techniques.
6. Assess the impact of sampling methods on the accuracy and precision of parameter estimation.

Learning Outcomes :-

1. Understanding the importance of sampling in statistical analysis.
2. Proficiency in implementing various sampling techniques in R.
3. Ability to estimate statistical parameters using different samples.
4. Comparing and interpreting the impact of sampling methods on parameters estimation accuracy.
5. Gaining insights into when to use specific sampling techniques based on the nature of the data.

11/4

Teacher's Signature:

```
> library(dplyr)
>
> # Load the iris dataset
> data("iris")
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

Environment History Connections Tutorial

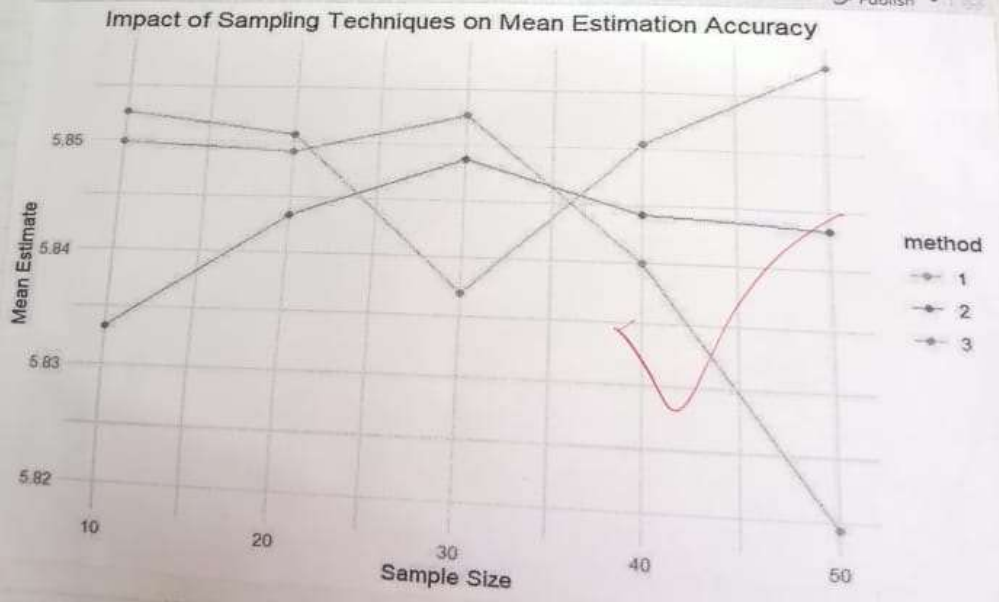
Import Dataset · 265 MiB · List

R · Global Environment

iris	150 obs. of 5 variables
Values	
num_iterations	100
sample_sizes	num [1:5] 10 20 30 40 50
sampling_methods	chr [1:3] "simple_random" "stratified" "systematic"
Functions	
assess_accuracy	function (dataset, sample_sizes, sampling_method, n
calculate_mean	function (vector)
perform_sampling	function (dataset, sample_size, sampling_method)

Files Plots Packages Help Viewer Presentation

Zoom · Export · Publish



SCREENSHOTS:

```r
# Load required libraries
library(dplyr)

# Load the iris dataset
data("iris")
head(iris)

# Function to calculate mean of a vector
calculate_mean <- function(vector) {
  return(mean(vector))
}

# Function to perform sampling and estimate mean
perform_sampling <- function(dataset, sample_size, sampling_method) {
  if (sampling_method == "simple_random") {
    sampled_data <- dataset[sample(nrow(dataset), sample_size, replace = FALSE), ]
  } else if (sampling_method == "stratified") {
    sampled_data <- dataset %>%
      group_by(Species) %>%
      sample_n(sample_size, replace = FALSE)
  } else if (sampling_method == "systematic") {
    # Selecting every nth observation
    n <- nrow(dataset)
    k <- sample_size
    start <- sample(1:k, 1)
    indices <- seq(start, n, by = k)
    sampled_data <- dataset[indices, ]
  } else {
    stop("Invalid sampling method.")
  }

  return(calculate_mean(sampled_data$Sepal.Length))
}

# Function to assess impact on accuracy of mean estimation
assess_accuracy <- function(dataset, sample_sizes, sampling_method, num_iterations) {
  results <- data.frame(sample_size = numeric(), mean_estimate = numeric())

  for (sample_size in sample_sizes) {
    mean_estimates <- replicate(num_iterations, perform_sampling(dataset, sample_size, sampling_method))
```

```
> library(dplyr)
>
> # Load the iris dataset
> data("iris")
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```r
    mean_estimate <- mean(mean_estimates)
    result <- data.frame(sample_size = sample_size, mean_estimate = mean_estima
    results <- rbind(results, result)
  }

  return(results)
}

# Define parameters
sample_sizes <- c(10, 20, 30, 40, 50)
num_iterations <- 100
sampling_methods <- c("simple_random", "stratified", "systematic")

# Assess accuracy for each sampling method
accuracy_results <- lapply(sampling_methods, function(method) {
  assess_accuracy(iris, sample_sizes, method, num_iterations)
})

# Plotting the results
library(ggplot2)
library(tidyr)

accuracy_df <- bind_rows(accuracy_results, .id = "method")
ggplot(accuracy_df, aes(x = sample_size, y = mean_estimate, color = method)) +
  geom_line() +
  geom_point() +
  labs(title = "Impact of Sampling Techniques on Mean Estimation Accuracy",
       x = "Sample Size", y = "Mean Estimate") +
  theme_minimal()
```

Environment | History | Connections | Tutorial

Import Dataset | 265 MiB | List

R | Global Environment

iris	150 obs. of 5 variables
Values	
num_iterations	100
sample_sizes	num [1:5] 10 20 30 40 50
sampling_methods	chr [1:3] "simple_random" "stratified" "systematic"
Functions	
assess_accuracy	function (dataset, sample_sizes, sampling_method, n...
calculate_mean	function (vector)
perform_sampling	function (dataset, sample_size, sampling_method)

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish