**# Experiment 1: Implementation of Data Charts**

# Load necessary library

library(ggplot2)

# Using built-in 'mtcars' dataset

data(mtcars)

# Bar Chart - Count of Cylinders

bar\_chart <- ggplot(mtcars, aes(x = factor(cyl))) +

geom\_bar(fill = "blue") +

labs(title = "Bar Chart of Cylinders", x = "Cylinders", y = "Count")

# Pie Chart - Proportion of Cylinders

cyl\_count <- table(mtcars$cyl)

pie(cyl\_count, labels = names(cyl\_count), main = "Pie Chart of Cylinders", col = rainbow(length(cyl\_count)))

# Line Chart - MPG vs Horsepower

line\_chart <- ggplot(mtcars, aes(x = hp, y = mpg)) +

geom\_line(color = "red") +

labs(title = "MPG vs Horsepower", x = "Horsepower", y = "Miles Per Gallon")

# Scatter Plot - Displacement vs MPG

scatter\_plot <- ggplot(mtcars, aes(x = disp, y = mpg)) +

geom\_point(color = "green") +

labs(title = "Displacement vs MPG", x = "Displacement", y = "Miles Per Gallon")

# Print plots

print(bar\_chart)

print(line\_chart)

print(scatter\_plot)

**# Experiment 2: Implementation of Data Visualization Techniques**

# Load necessary library

library(ggplot2)

# Using built-in 'iris' dataset

data(iris)

# Histogram - Distribution of Sepal Length

histogram\_plot <- ggplot(iris, aes(x = Sepal.Length)) +

geom\_histogram(fill = "blue", bins = 15, color = "black") +

labs(title = "Histogram of Sepal Length", x = "Sepal Length", y = "Frequency")

# Box Plot - Sepal Width per Species

box\_plot <- ggplot(iris, aes(x = Species, y = Sepal.Width, fill = Species)) +

geom\_boxplot() +

labs(title = "Box Plot of Sepal Width by Species", x = "Species", y = "Sepal Width")

# Grouped Bar Plot - Mean Petal Length per Species

library(dplyr)

mean\_petal <- iris %>% group\_by(Species) %>% summarize(mean\_petal\_length = mean(Petal.Length))

grouped\_bar\_plot <- ggplot(mean\_petal, aes(x = Species, y = mean\_petal\_length, fill = Species)) +

geom\_bar(stat = "identity") +

labs(title = "Mean Petal Length by Species", x = "Species", y = "Mean Petal Length")

# Line Plot - Sepal Length vs Sepal Width

line\_plot <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +

geom\_line() +

labs(title = "Sepal Length vs Sepal Width", x = "Sepal Length", y = "Sepal Width")

# Print plots

print(histogram\_plot)

print(box\_plot)

print(grouped\_bar\_plot)

print(line\_plot)

Experiment 3: Designing Multivariate Patterns

# Load necessary libraries

library(ggplot2)

library(GGally)

library(reshape2)

# Using built-in 'mtcars' dataset

data(mtcars)

# Scatterplot Matrix - Visualizing relationships between multiple variables

scatter\_matrix <- ggpairs(mtcars, columns = c("mpg", "disp", "hp", "wt", "qsec"))

# Heat Map - Correlation between numeric variables

mtcars\_corr <- round(cor(mtcars), 2)

melted\_corr <- melt(mtcars\_corr)

heat\_map <- ggplot(melted\_corr, aes(Var1, Var2, fill = value)) +

geom\_tile() +

scale\_fill\_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +

labs(title = "Heat Map of Correlation", x = "Variables", y = "Variables") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# Parallel Coordinates Plot - Visualizing multiple variables simultaneously

library(MASS)

data(iris)

parallel\_plot <- ggparcoord(iris, columns = 1:4, groupColumn = 5, alphaLines = 0.5) +

labs(title = "Parallel Coordinates Plot for Iris Dataset")

# Print plots

print(scatter\_matrix)

print(heat\_map)

print(parallel\_plot)

**# Experiment 3: Designing Multivariate Patterns**

# Load necessary libraries

library(ggplot2)

library(GGally)

library(reshape2)

# Using built-in 'mtcars' dataset

data(mtcars)

# Scatterplot Matrix - Visualizing relationships between multiple variables

scatter\_matrix <- ggpairs(mtcars, columns = c("mpg", "disp", "hp", "wt", "qsec"))

# Heat Map - Correlation between numeric variables

mtcars\_corr <- round(cor(mtcars), 2)

melted\_corr <- melt(mtcars\_corr)

heat\_map <- ggplot(melted\_corr, aes(Var1, Var2, fill = value)) +

geom\_tile() +

scale\_fill\_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +

labs(title = "Heat Map of Correlation", x = "Variables", y = "Variables") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# Parallel Coordinates Plot - Visualizing multiple variables simultaneously

library(MASS)

data(iris)

parallel\_plot <- ggparcoord(iris, columns = 1:4, groupColumn = 5, alphaLines = 0.5) +

labs(title = "Parallel Coordinates Plot for Iris Dataset")

# Print plots

print(scatter\_matrix)

print(heat\_map)

print(parallel\_plot)

**# Experiment 4: Understand and implement visual encoding of data using R**

# Load required libraries

library(ggplot2)

# Load built-in dataset

data(mtcars)

# Convert 'cyl' (cylinders) into a factor for categorical encoding

mtcars$cyl <- as.factor(mtcars$cyl)

# View first few rows of the dataset

head(mtcars)

#1️⃣ Distribution of a Continuous Variable (MPG)

ggplot(mtcars, aes(x = mpg)) +

geom\_histogram(binwidth = 2, fill = "skyblue", color = "black") +

labs(title = "Distribution of MPG", x = "Miles Per Gallon (MPG)", y = "Count") +

theme\_minimal()

# 2️⃣ Relationship Between Two Continuous Variables (MPG vs Weight)

ggplot(mtcars, aes(x = wt, y = mpg, color = cyl)) +

geom\_point(size = 3) +

labs(title = "Scatter Plot: Weight vs MPG", x = "Weight (1000 lbs)", y = "Miles Per Gallon (MPG)") +

theme\_minimal()

# 3️⃣ Distribution of a Categorical Variable (Number of Cylinders)

ggplot(mtcars, aes(x = cyl, fill = cyl)) +

geom\_bar() +

labs(title = "Bar Chart: Count of Cars by Cylinders", x = "Number of Cylinders", y = "Count") +

theme\_minimal()

# 4️⃣ Relationship Between Continuous and Categorical Variables (Horsepower by Cylinders)

ggplot(mtcars, aes(x = cyl, y = hp, fill = cyl)) +

geom\_boxplot() +

labs(title = "Box Plot: Horsepower by Cylinder Count", x = "Number of Cylinders", y = "Horsepower") +

theme\_minimal()

**# Experiment 5: Dashboard Designing using Shiny**

# Load necessary libraries

library(shiny)

library(ggplot2)

# Using built-in 'mtcars' dataset

data(mtcars)

# Define UI for the dashboard

ui <- fluidPage(

titlePanel("Car Data Dashboard"),

sidebarLayout(

sidebarPanel(

selectInput("var", "Choose a variable:", choices = names(mtcars), selected = "mpg")

),

mainPanel(

plotOutput("histPlot")

)

)

)

# Define server logic

server <- function(input, output) {

output$histPlot <- renderPlot({

ggplot(mtcars, aes\_string(x = input$var)) +

geom\_histogram(fill = "blue", bins = 10, color = "black") +

labs(title = paste("Histogram of", input$var), x = input$var, y = "Frequency")

})

}

# Run the application

shinyApp(ui = ui, server = server)

**# Experiment 6: Multivariate Displays**

# Load necessary libraries

library(ggplot2)

library(GGally)

library(plotly)

# Using built-in 'iris' dataset

data(iris)

# Faceted Scatter Plot - Petal Length vs Petal Width grouped by Species

faceted\_scatter <- ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +

geom\_point() +

facet\_wrap(~Species) +

labs(title = "Faceted Scatter Plot: Petal Length vs Petal Width", x = "Petal Length", y = "Petal Width")

# 3D Scatter Plot - Sepal Length, Sepal Width, and Petal Length

plot\_3d <- plot\_ly(iris, x = ~Sepal.Length, y = ~Sepal.Width, z = ~Petal.Length, color = ~Species, type = "scatter3d", mode = "markers")

# Pair Plot - Multivariate relationships

pair\_plot <- ggpairs(iris, columns = 1:4, mapping = aes(color = Species))

# Print plots

print(faceted\_scatter)

print(pair\_plot)

plot\_3d

**# Experiment 7: Implementation of Overplotting Solutions**

# Load necessary library

library(ggplot2)

# Using built-in 'mtcars' dataset

data(mtcars)

# Basic Scatter Plot - Weight vs MPG (Shows Overplotting)

overplot\_scatter <- ggplot(mtcars, aes(x = wt, y = mpg)) +

geom\_point(alpha = 1, color = "red") +

labs(title = "Overplotted Scatter Plot: Weight vs MPG", x = "Weight", y = "Miles Per Gallon")

# Solution 1: Transparency (alpha)

scatter\_alpha <- ggplot(mtcars, aes(x = wt, y = mpg)) +

geom\_point(alpha = 0.4, color = "blue") +

labs(title = "Scatter Plot with Transparency", x = "Weight", y = "Miles Per Gallon")

# Solution 2: Jittering

scatter\_jitter <- ggplot(mtcars, aes(x = wt, y = mpg)) +

geom\_jitter(alpha = 0.6, color = "green") +

labs(title = "Scatter Plot with Jittering", x = "Weight", y = "Miles Per Gallon")

# Solution 3: Density Plot (alternative to scatter)

density\_plot <- ggplot(mtcars, aes(x = wt, y = mpg)) +

geom\_bin2d() +

scale\_fill\_gradient(low = "blue", high = "red") +

labs(title = "Density Plot: Weight vs MPG", x = "Weight", y = "Miles Per Gallon")

# Print plots

print(overplot\_scatter)

print(scatter\_alpha)

print(scatter\_jitter)

print(density\_plot)