Program for Data Visualization: Use packages like ggplot2 or plot to create various types of charts, such as bar charts, line plots, scatter plots, or heatmaps.

• Read a dataset from a CSV file or other data sources.

• Customise the charts by adding labels, titles, legends, and adjusting the axis scales.

• Create interactive visualizations with tooltips, zooming, or filtering options.

• Export the visualizations to different file formats or display them within an R notebook or Shiny application.

# Load necessary packages

library(ggplot2)

library(dplyr)

library(plotly)

# Step 1: Read Titanic dataset

titanic <- read.csv("train.csv")

# Convert factor levels

titanic$Survived <- factor(titanic$Survived, levels = c("0", "1"))

# Step 2: Customise the charts

# Bar chart showing count of survivors by passenger class

bar\_chart <- ggplot(titanic, aes(x = factor(Pclass), fill = factor(Survived))) +

geom\_bar(position = "dodge") +

labs(title = "Survivors by Passenger Class",

x = "Passenger Class",

y = "Count",

fill = "Survived") +

scale\_fill\_manual(values = c("0" = "red", "1" = "blue")) # Ensure consistent levels with the factor

# Line plot showing age distribution of passengers

line\_plot <- ggplot(titanic, aes(x = Age, y = ..density..)) +

geom\_density(fill = "blue", alpha = 0.5) +

labs(title = "Age Distribution of Passengers",

x = "Age",

y = "Density")

# Scatter plot showing fare vs age with color indicating survival status

scatter\_plot <- ggplot(titanic, aes(x = Age, y = Fare, color = factor(Survived))) +

geom\_point() +

labs(title = "Fare vs Age",

x = "Age",

y = "Fare",

color = "Survived") +

scale\_color\_manual(values = c("0" = "red", "1" = "blue")) # Ensure consistent levels with the factor

#heatmap

heatmap <- ggplot(titanic, aes(x = Pclass, y = Survived)) +

stat\_bin\_2d(bins = 10, aes(fill = ..count..)) +

labs(title = "Titanic Survival Heatmap", x = "Pclass", y = "Survived", fill = "Frequency") +

scale\_fill\_continuous(name = "Frequency", low = "white", high = "blue") +

theme\_minimal()

# Step 3: Create interactive visualizations using plotly

# Interactive bar chart

interactive\_bar\_chart <- ggplotly(bar\_chart)

# Interactive line plot

interactive\_line\_plot <- ggplotly(line\_plot)

# Interactive scatter plot

interactive\_scatter\_plot <- ggplotly(scatter\_plot)

#interactive heat map

interactive\_heatr\_plot <- ggplotly(heatmap)

# Step 4: Export interactive visualizations

# Export visualizations to png files

ggsave("bar\_plot.png", plot = bar\_plot, width = 8, height = 6)

ggsave("line\_plot.png", plot = line\_plot, width = 8, height = 6)

ggsave("scatter\_plot.png", plot = scatter\_plot, width = 8, height = 6)

ggsave("titanic\_heatmap\_ggplot.png", heatmap, width = 8, height = 6)

# Export plotly interactive visualizations to HTML files

htmlwidgets::saveWidget(interactive\_bar\_chart, "interactive\_bar\_chart.html", selfcontained = TRUE)

htmlwidgets::saveWidget(interactive\_line\_plot, "interactive\_line\_plot.html", selfcontained = TRUE)

htmlwidgets::saveWidget(interactive\_scatter\_plot, "interactive\_scatter\_plot.html", selfcontained = TRUE)

htmlwidgets::saveWidget(interactive\_heatr\_plot, file = "titanic\_heatmap.html")