DATA VISUALIZATION USING SHINY APP

TITANIC DATASET

Team 6:

Hamsini Reddy Gunda

Sowmya Harini Devi Kumpatla

Elisha Narimalla

Sri Harsha Adapa

Sai Krishna Davuluri

CONTENTS

I. INTRODUCTION:

In data science, the Titanic data set is known for containing detailed information about the ship's passengers during her deadly first journey. With its diverse set of characteristics, including age, class, price, and survivor status, this dataset provides an excellent opportunity for experimental data analysis (EDA). The significance of EDA comes from its ability to show insights, patterns, and trends concealed within the dataset, resulting in a more complete understanding of social and economic processes and variables influencing survival rates.

In this project, we use Shiny which is an interactive R web application system to create a dynamic and user-friendly Titanic dataset display tool. With Shiny, we can simply transform static data into an engaging and intriguing experience that allows visitors to explore the dataset's inner elements.

II. OBJECTIVES:

The Shiny app portrays the Titanic dataset with specific goals to provide an engaging and instructive experience for users. The following is a summary of the primary objectives:

- **Easy-to-use:** Provide a user-friendly interface for exploring the Titanic dataset, regardless of skill level.
- **Justification:** Improve accessibility and ease of use to allow users to simply browse the app and gain crucial insights.
- **Variable Comparison:** Allow users to dynamically select and compare variables such as age, gender, class, and price.
- **Justification:** Allow users to investigate how different factors influence rates of survival and other dataset characteristics.
- **Age details:** Provide a slider or entry method to set age ranges for analysis.
- **Plot types:** Use several plot types, including pie charts, violin plots, scatter plots, three-dimensional scatter plots, and others. Give an in-depth review of the dataset to accommodate different analytical needs and preferences.
- **Plot Generation in Real Time:** Use server-side logic to produce dynamic charts based on age ranges, plot styles, and user-selected factors. Give users fast feedback so they can observe how changes to input parameters affect the visualizations.

III. METHODOLOGY:

This section outlines the iterative and structured approach in the methods and processes used to visualize the Titanic dataset by creating an interactive and informative Shiny app. The following are the actions or methods taken for visualizing Titanic dataset:

1. **Data preprocessing**: Preprocessing operations are performed on the dataset, including data type conversion, missing value correction, and data integrity checking. This is important since precise visualizations and accurate representations depend on well-organized and correct data.
2. **Install all the necessary libraries:** Important R packages like ggplot2, shiny, plotly and tidy verse are installed to facilitate the creation of Shiny app and data visualization. These packages provide a lot of features that can be used to create accurate data visualizations and construct interactive applications.
3. **Structural design of Shiny app:** The structural design of the Shiny app delineates the clear separation between server-side functionality (back-end) and user interface (UI) elements (front-end). The server logic manages tasks such as generating plots and implementing various functionalities like setting parameters such as number of bins, fill, color, etc. On the other hand, the UI includes features like application title, side bar with slider input for adjusting the number of bins, etc. Organizing the application in this manner enhances readability and simplifies maintenance of the code.
4. **Plot Type Integration:** The dataset is visualized using different plots like scatter plot, box plot, bar chart, jitter plot, point plot, etc., The program integrates much more diverse types of plots, such as pie charts, scatter plots, violin plots, 3D scatter plots, and more. This diverse range of display styles offers a comprehensive overview of the dataset and caters to different analytical requirements.

IV. SHINY APP OVERVIEW:

Users can easily browse and examine various parts of the data through a Shiny app, which is used to represent the Titanic dataset visually. Below is a summary of the key features and components of the app:

1. Selecting a Plot Type: The user can choose from a wide spectrum of plot options that may be used in all fields of statistics and usually represent bubble plots, violin plots, scatter

plots, 3D scatter plots, pie charts, frequency polygons, histograms, density plots, bar plots, jitter plots, bubble plots, and point plots. Here the default plot type is "Scatter Plot."

2. Data Filtering Options: Users can select predefined choices for the x and y axes, such as Age, Passenger Id, Pclass, SibSp, Parch, and Fare, or "None" for certain plot kinds. The default values for the x- and y-axes are "Age" and "Fare", respectively.
3. Age Range Details: Users will be able to set an age consistency by using the slider control. The dataset is filtered for a more focused exploration based on the chosen age range.
4. The Main Panel: It displays the plot type selected by the user and dynamically updates in real time when modifications are made.
5. Plot Output: The plot output, named "selected Plot," which is created based on the selected plot type, age range, and x- and y-axis variables. These plots utilize color aesthetics to distinguish between male and female passengers.
6. Plot Types: Numerous plot types are offered by the app, which makes every point different based on Titanic data. For instance, Scatter plots are utilized to plot the independence of two variables. Density and Histogram plots show the distribution of a single variable. Jitter plots are used to represent the distribution of a single variable while also displaying individual data points, allowing for the identification of patterns or clusters within the data. Plots like Box plots and Violin plots show how a variable is spread across multiple categories. 3D scatter plots are used for the visualization of data points in three dimensions.
7. Interactive Elements: The application combines interactivity modules like zoom in/out and pop-up details to give users more information and a very education-oriented exploration experience.
8. Integration of Plots: "3D Scatter Plot" plot type allows the users to interact with the 3D model of plot based on converting the ggplot object into a plotly object through the ggplot function.
9. Customization Features: This application contains features for modifying the appearance of plots, such as changing colors, labels, and titles, to better suit the user's preferences and analytical needs.
10. Export and Sharing Options: Users can export plots or download the underlying data for further study. Likewise, sharing options may also make it easier for users to share their visualizations with others and collaborate on projects.

V. PLOT TYPES:

Users using the app will be able to use the Shiny app you provided to explore completely different plot types based on the Titanic dataset. The plots that we are using in this Shiny App are:

1. Scatter Plot: A picture showing individual points' data twins, with two variables on the graph's x and y axes, illustrating the relationship or connection between them.
2. Line Plot: A line graph that uses a certain convention to plot the data points that follow one another and connect by straight lines is often used to show a change that happens over time or continuous data.
3. Box Plot: It is sometimes called a box-and-whisker plot. It shows visually the distribution of data through quartiles, drawn upon the median, between the interquartile range, and with the outliers included if present.
4. Frequency Polygon: A graph that utilizes one of the many forms of the line chart, which plots the frequency of various values on the x-axis and corresponding values on the y-axis. This structure forms a polygon when connected with straight lines.
5. Histogram: The representation of a graphical distribution of numerical data, whereby data is divided into intervals (bins), and the height of each bar signifies the frequency of data points falling into that interval.
6. Density Plot: A smooth version of a graph that portrays the distribution of data as a continuous slope or curve is typically used to represent the probability density function of a continuous variable.
7. Bar Plot: The visual representation of categorical data using rectangular bars to clearly show a direct relationship between the length and width of bars and the value of the data represented is often used when comparing values in various categories.
8. Jitter Plot: With a scatter plot, the data should be jittered, so the individual points do not overlap with each other; hence, a cleaner representation of the distribution is achieved.
9. Violin Plot: The categorical distribution is shown across different categories using a box plot, and the density plot of the symmetrical kernel density is plotted mirrored along the y-axis.

10. Bubble Plot: This is a flu variation of a scatter chart, where the points are depicted as bubbles, and the size of each bubble reflects a third variable while holding the other two constants.
11. Point Plot: A plot that is usually used to show probability point estimates and confidence intervals, as well as in statistical analysis and data visualization.

VI. CODE:

```
# Load necessary libraries

library(shiny)

library(tidyverse)

library(ggplot2)

library(plotly)

# Read data

data <- read.csv ('/Users/Download/Project/sowmy/train.csv')

# Define UI

ui <- fluidPage(

 fluidRow(

  column(

   width = 6,

   selectInput(

    "plotType",

    "Select Plot Type:",

    choices = c(

     "Scatter Plot", "Line Plot", "Box Plot", "FreqPoly",
```

```
        "Histogram Plot", "Density Plot", "Bar Plot", "Jitter Plot", "Violin Plot",
        "Bubble Plot", "Point Plot"
      ),
      selected = "Scatter Plot"
    )
  )
),
selectInput(
  "xVariable",
  "Select x-axis variable:",
  choices = c("Age", "PassengerId", "Pclass", "SibSp", "Parch", "Fare", "None"),
  selected = "Age"
),
selectInput(
  "yVariable",
  "Select y-axis variable:",
  choices = c("Age", "PassengerId", "Pclass", "SibSp", "Parch", "Fare"),
  selected = "Fare"
),
# Add a slider input for Age
sliderInput(
  "ageRange", "Select Age Range:",
  min = min(data$Age, na.rm = TRUE), max = max(data$Age, na.rm = TRUE),
```

```
    value = c(min(data$Age, na.rm = TRUE), max(data$Age, na.rm = TRUE))
  ),
  mainPanel(
    fluidRow(
      column(width = 10, plotlyOutput(outputId = "selectedPlot"))
    )
  )
)
# Define server logic
server <- function(input, output) {
  output$selectedPlot <- renderPlotly({
    plotType <- input$plotType
    xVariable <- input$xVariable
    yVariable <- input$yVariable
    # Filter data based on the selected age range
    filtered_data <- data %>%
      filter(Age >= input$ageRange[1], Age <= input$ageRange[2])
    p <- switch(
      plotType,
      "Scatter Plot" = {
        ggplot(filtered_data, aes_string(x = xVariable, y = yVariable, color = "Sex")) +
          geom_point() +
          labs(x = xVariable, y = yVariable, title = "Scatter Plot colored by Sex") +
```

```
    theme_minimal()
  },
  "Line Plot" = {
    ggplot(filtered_data, aes_string(x = xVariable, y = yVariable, color = "Sex")) +
      geom_line() +
      labs(x = xVariable, y = yVariable, title = "Line Plot colored by Sex") +
      theme_minimal()
  },
  "Box Plot" = {
    ggplot(filtered_data, aes_string(x = xVariable, y = yVariable, color = "Sex")) +
      geom_boxplot() +
      labs(x = xVariable, y = yVariable, title = "Box Plot colored by Sex") +
      theme_minimal()
  },
  "FreqPoly" = {
    ggplot(filtered_data, aes(x = .data[[xVariable]], color = Sex)) +
      geom_freqpoly(binwidth = 5) +
      labs(x = xVariable, title = "FreqPoly Plot") +
      theme_minimal()
  },
  "Histogram Plot" = {
    if ("Sex" %in% colnames(filtered_data)) {
      ggplot(filtered_data, aes_string(x = xVariable, fill = "Sex")) +
```

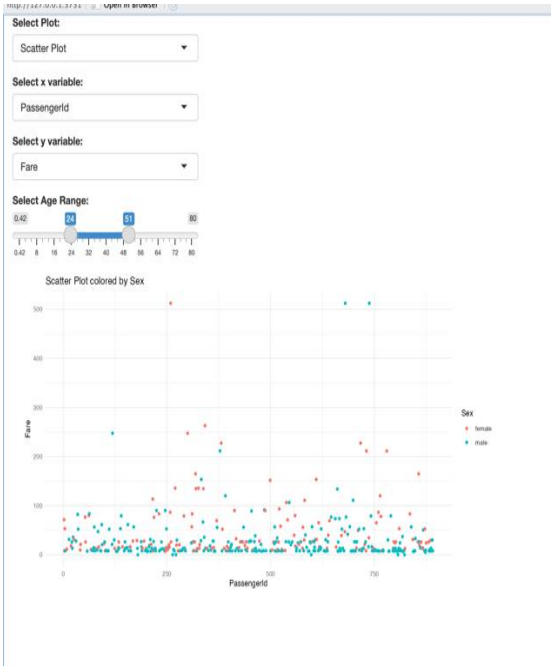
```
      geom_histogram(binwidth = 10) +
      labs(x = xVariable, title = paste("Histogram Plot", "colored by Sex")) +
      theme_minimal()
    } else {
     ggplot(filtered_data, aes_string(x = xVariable)) +
      geom_histogram(binwidth = 10) +
      labs(x = xVariable, title = "Histogram Plot") +
      theme_minimal()
    }
  },
  "Density Plot" = {
    ggplot(filtered_data, aes_string(x = xVariable, fill = "Sex")) +
     geom_density() +
     labs(x = xVariable, title = "Density Plot") +
     theme_minimal()
  },
  "Bar Plot" = {
    ggplot(filtered_data, aes_string(x = xVariable, fill = "Sex")) +
     geom_bar(stat = "count") +
     labs(x = xVariable, title = "Bar Plot") +
     theme_minimal()
  },
  "Jitter Plot" = {
```

```
      ggplot(filtered_data, aes_string(x = xVariable, y = yVariable, color = "Sex")) +
        geom_jitter(position = position_jitter(0.2)) +
        labs(x = xVariable, title = "Jitter Plot") +
        theme_minimal()
    },
    "Violin Plot" = {
      ggplot(filtered_data, aes_string(x = xVariable, y = yVariable, fill = "Sex")) +
        geom_violin() +
        labs(x = xVariable, title = "Violin Plot") +
        theme_minimal()
    },
    "Bubble Plot" = {
      ggplot(filtered_data, aes_string(x = xVariable, y = yVariable, size = "Age", color =
"Sex")) +
        geom_point() +
        labs(x = xVariable, y = "Age", title = "Bubble Plot") +
        theme_minimal()
    },
    "Point Plot" = {
      ggplot(filtered_data, aes_string(x = xVariable, y = yVariable, color = "Sex")) +
        geom_point() +
        labs(x = xVariable, title = "Point Plot") +
        theme_minimal()
```

```
    }
  )
 })
}
# Run the application
shinyApp(ui = ui, server = server)
```

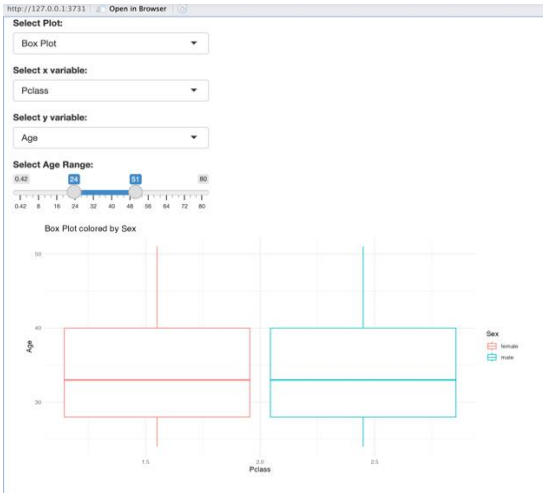
VII. RESULTS:

Scatter Plot:

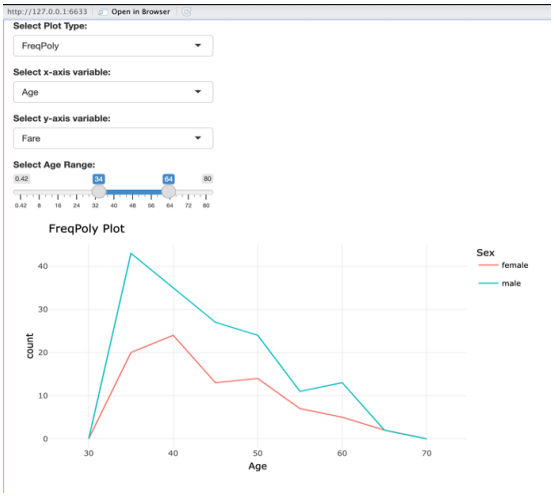


Line Plot:

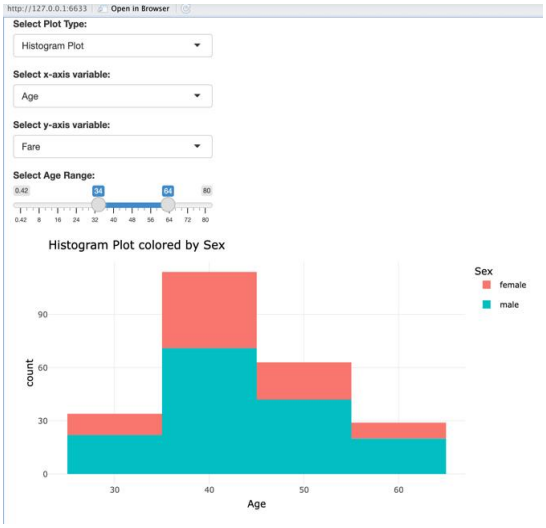
Box Plot:



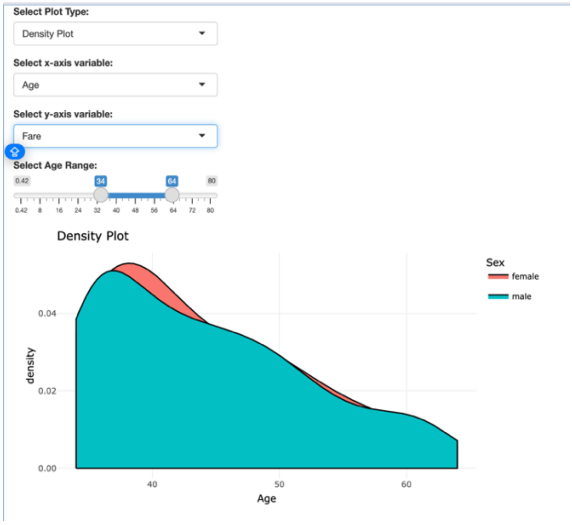
FreqPloy:



Histogram:



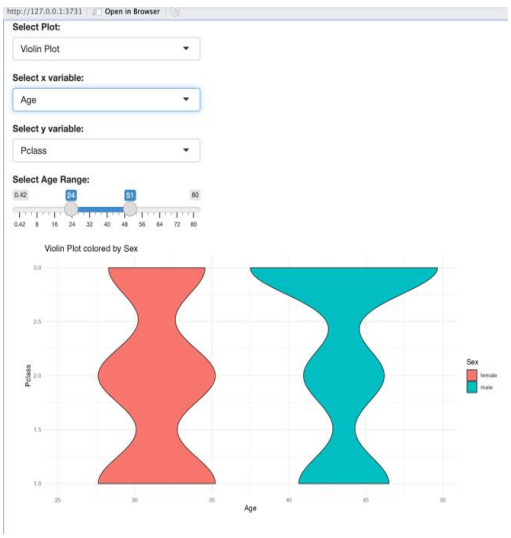
Density Plot:



Bar Plot:

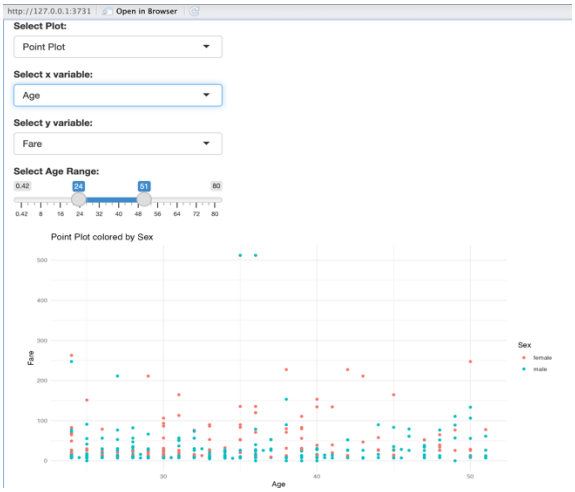
Jitter Plot:

Violin Plot:



Bubble Plot:

Point Plot:



VIII. DISCUSSIONS:

1. The significance of exploratory data analysis (EDA)

   Significance: EDA is a critical step in data analysis where analysts investigate and summarize important attributes of the data and try to understand trends, identify outliers, and identify hypotheses for future exploration.

2. User Communication and Discovery

   Shiny App Design: The design of shiny app is based on the creation of web-based applications, which should allow communication between data and users. It gives users an opportunity to intuitively explore and visualize data, helping in discovery and understanding.

3. Analysis of Demographics

   Gender Distribution: This is done by analyzing the gender distribution between two different groups or samples to know the gender makeup and its inequality within the target population being studied.

4. Trends Associated with Age

   Age Range Slider: This analysis includes analyzing the trends or patterns associated with people of different ages. The age range slider allows users to get more into the details by filtering information via certain age ranges.

5. Analysis of Survival

   Color-Coded Survival Rates: This research aims at comparing survival rates among a population or a sample and these conclusions can be presented in a graphical form using different colors to stress the survival rate variations among different groups or conditions.

IX. CONCLUSION:

The use of the Shiny app for exploring the Titanic dataset through interactive visualizations has uncovered valuable insights across different aspects such as survival trends, demographics, and correlations. Here's a breakdown of the key findings:

Demographic Patterns: The visualizations underscored the gender distribution aboard the Titanic, offering a clear view of the ratio between male and female passengers.

Age Trends: Visual representations, including density plots and histograms, effectively illustrated the age distribution among passengers. By pinpointing peaks and trends within various age brackets, a nuanced understanding of passenger demographics was achieved.

Survival Analysis: Color-coded plots such as box and violin plots enabled a thorough examination of factors influencing survival rates. These graphics effectively demonstrated how certain variables impacted the likelihood of survival.

Class Disparities: Visualizations based on passenger class (Pclass) shed light on class-related trends, showcasing how different classes influenced factors like age, fare, and survival rates.

Multivariate Exploration: The incorporation of sophisticated visualizations like bubble plots and three-dimensional scatter plots allowed for the simultaneous exploration of three variables. This multivariate approach provided a comprehensive perspective on the dataset, enhancing the depth of analysis.