Implementation Report: Road Traffic Accident (RTA) Analysis Using Shiny App

Submitted by: AS-Saba

For: Prof. Dominik Bohler
Program: Digital Health – Master of Science in Global Public Health

Deliverables:

- GitHub Repository
- Shiny Application

Submitted on: 28.01.2026

1. Problem Statement

Road Traffic Accidents (RTA) are a leading cause of injuries and fatalities globally, and Pakistan has one of the highest rates of road accidents in South Asia. The lack of proper analysis and visualization of RTA data prevents policymakers, road safety authorities, and public health professionals from designing effective interventions.

Introduction

Road traffic accidents (RTA) are a major public health concern worldwide, contributing significantly to injury, disability, and mortality. According to the World Health Organization, approximately 1.3 million people die each year due to road accidents, and millions more suffer from non-fatal injuries. In countries like Pakistan, the problem is further compounded by high traffic density, insufficient road safety infrastructure, and limited awareness about safe driving practices.

Understanding the factors contributing to accident severity—such as helmet use, driving experience, vehicle type, area (urban/rural), and driver demographics—is crucial for designing effective interventions. Traditional static data reports often fail to provide actionable insights, making it difficult for policymakers and public health professionals to identify high-risk areas and behaviors.

Interactive tools, like Shiny apps in R, offer a dynamic approach to visualize and explore accident data, allowing users to filter, compare, and analyze multiple variables simultaneously. This project leverages these capabilities to create a platform that aids in identifying risk factors and informing preventive strategies

Problem Statement

Pakistan faces one of the highest rates of road traffic accidents in South Asia, resulting in significant social and economic losses. Despite the availability of accident records, there is a **lack of interactive tools** that can:

- Identify the **key factors contributing to accident severity** (behavioral, demographic, and environmental).
- Compare trends across **urban and rural areas**, gender, and other demographics.
- Provide **visual insights** that can guide policy-making and awareness campaigns.

The challenge lies in transforming raw accident data into **actionable insights** that can inform road safety measures, reduce severe injuries, and ultimately save lives.

Objective: To develop a Shiny app that dynamically visualizes RTA data, identifies high-risk factors, and facilitates data-driven interventions in road safety.

Solution Pathway (Project-Based Visualizations)

The RTA Shiny app transforms raw accident data into **easy-to-understand patterns** that can be used to create **awareness, prevent accidents, and guide policy decisions**. Each graph targets specific factors contributing to severe injuries.

1. Severity by Area (Grouped / Stacked Bar Chart)

Illustration:
This chart compares accident severity (minor vs severe) across different **areas**, including residential, office, market, school, rural villages, and other locations. Bars are colored by severity and grouped for each area to highlight differences. Interactive tooltips display exact counts and percentages.

Insight:

- Rural village roads and busy market areas show **higher proportions of severe injuries**.
- Urban residential or office zones tend to have more minor accidents, likely due to better road infrastructure and quicker emergency response.

Lesson / Action Point:

- Prioritize **road safety improvements** in high-risk zones (e.g., signage, traffic calming, lighting).
- Run targeted **awareness campaigns** for pedestrians and drivers in these areas.
- Focus emergency services on areas with historically high severe accident rates.

2. Severe Injuries by Day (Line / Time Series Chart)

Illustration:
This graph shows how **severe injuries vary across the days of the week**. A line or point plot highlights peaks and valleys, allowing users to identify **temporal trends**.

Insight:

- Certain days, especially weekends, show **more severe injuries**, possibly due to higher traffic, late-night travel, or risky behavior.

Lesson / Action Point:

- Increase **traffic enforcement** and **police patrolling** during high-risk days.
- Plan **road safety awareness campaigns** before weekends.
- Introduce **driver behavior monitoring programs** during peak accident periods

3. Vehicle Type vs Severity (Bar Chart)

Illustration:
This visualization compares accident severity across **vehicle types**, such as motorcycles, cars, trucks, buses, and bicycles. Bars are colored to indicate severity level, and interactive tooltips provide exact counts.

Insight:

- Motorcycles have the **highest proportion of severe injuries** due to exposure and lack of protective structures.
- Overloaded trucks or buses are also linked with more severe outcomes.

Lesson / Action Point:

- Implement **vehicle-specific safety measures** (helmets, protective gear, load limits).
- Conduct **driver training programs** for commercial vehicle operators.
- Raise **public awareness** about the risks of two-wheelers and overloaded vehicles.

4. Multi-Factor Interactive Analysis (Shiny Filters)

Illustration:
Users can dynamically filter data by **Area, Gender, Helmet Use**, and select X/Y variables to generate interactive plots. Tooltips show precise statistics for each combination.

Insight:

- Multi-factor analysis reveals **hidden high-risk groups**, such as **rural male motorcyclists without helmets**.
- Allows evaluation of combined effects of **behavioral, environmental, and demographic factors**.

Lesson / Action Point:

- Design **targeted interventions** for groups at the highest risk.
- Combine infrastructure improvements with behavioral campaigns for maximum impact.
- Use interactive insights to **inform policy decisions and prioritize resources**.

RTA Project: Graphs Code Workflow

1. Data Preparation

```
# Load libraries
library(dplyr)
library(ggplot2)
library(plotly)
library(readr)

# Load the dataset
df <- read_csv("RTA_dataset.csv")

# Clean column names
df_clean <- df %>%
  rename_all(~gsub(" ", "_", .)) %>%  # replace spaces with _
  mutate(
    Area = as.factor(Area),
    Severity = as.factor(Severity),
    Gender = as.factor(Gender),
    Vehicle_type = as.factor(Vehicle_type),
    Helmet_use = as.factor(Helmet_use),
```

```
    Day = as.factor(Day),
    Driving_experience = as.factor(Driving_experience)
  )

# Check for missing values
colSums(is.na(df_clean))
```

Explanation:

- Converts categorical variables to factors for plotting.
- Removes spaces from column names for easy referencing.
- df_clean is your main working dataframe.

2. Binning / Aggregation for Plots

For many of your graphs, you **binned or grouped the data** to calculate percentages of severe vs non-severe injuries.

```
# Example: Percent Severe by Area
df_binned <- df_clean %>%
  group_by(Area, Severity) %>%
  summarise(Count = n()) %>%
  ungroup() %>%
  group_by(Area) %>%
  mutate(
    Total = sum(Count),
    Percent = Count / Total * 100
  ) %>%
  ungroup()
```

Explanation:

- group_by(Area, Severity) groups data by area and severity.
- Count = n() counts number of accidents in each group.
- Percent calculates the **proportion of each severity** per area.

3. Optional: Standard Error / Uncertainty

If you want error bars:

```
df_binned <- df_binned %>%
  mutate(
    SE = sqrt((Percent/100)*(1-(Percent/100))/Total) * 100
  )
```

Explanation:

- Standard error of a proportion: sqrt(p*(1-p)/n).
- Multiplied by 100 to convert to percentage units.

4. Graph Construction

4.1 Severity by Area (Stacked Bar Chart)

```
p_area <- ggplot(df_binned, aes(x=Area, y=Percent, fill=Severity)) +
  geom_bar(stat="identity", position="stack") +
  scale_fill_manual(values=c("Minor"="green", "Severe"="red", "Fatal"="black")) +
  theme_minimal() +
  labs(
    title="Severity of Accidents by Area",
    x="Area",
    y="Percentage of Accidents",
    fill="Severity"
  )

# Show plot
p_area
```

Interpretation:

- Stacked bars show how **minor, severe, and fatal accidents distribute across areas**.
- Red section highlights the high-risk areas.

4.2 Severe Injuries by Day (Line / Trend Plot)

```
df_day <- df_clean %>%
  filter(Severity == "Severe") %>%
  group_by(Day) %>%
  summarise(Severe_Count = n()) %>%
  ungroup()

p_day <- ggplot(df_day, aes(x=Day, y=Severe_Count, group=1)) +
  geom_line(color="red", size=1.2) +
  geom_point(color="red", size=3) +
  theme_minimal() +
  labs(
    title="Severe Injuries by Day of the Week",
    x="Day",
    y="Number of Severe Injuries"
  )

p_day
```

Interpretation:

- Shows **temporal trends** in severe injuries.
- Peaks indicate high-risk days (e.g., weekends).

4.3 Vehicle Type vs Severity (Grouped Bar Chart)

```
df_vehicle <- df_clean %>%
  group_by(Vehicle_type, Severity) %>%
  summarise(Count = n()) %>%
  ungroup() %>%
  group_by(Vehicle_type) %>%
  mutate(
    Total = sum(Count),
    Percent = Count / Total * 100
  ) %>%
  ungroup()

p_vehicle <- ggplot(df_vehicle, aes(x=Vehicle_type, y=Percent, fill=Severity)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_manual(values=c("Minor"="green", "Severe"="red", "Fatal"="black")) +
  theme_minimal() +
  labs(
    title="Accident Severity by Vehicle Type",
    x="Vehicle Type",
    y="Percentage of Accidents",
    fill="Severity"
  )

p_vehicle
```

Interpretation:

- Compares severity proportions across **vehicle types**.
- Motorcycles and heavy vehicles usually show **higher severe accident percentages**.

4.4 Helmet Use vs Severe Injury (Grouped Bar Chart)

```
df_helmet <- df_clean %>%
  group_by(Helmet_use, Severity, Area) %>%
  summarise(Count = n()) %>%
  ungroup() %>%
  group_by(Helmet_use, Area) %>%
  mutate(
    Total = sum(Count),
    Percent = Count / Total * 100
```

```
  ) %>%
  ungroup()

p_helmet <- ggplot(df_helmet, aes(x=Helmet_use, y=Percent, fill=Severity)) +
  geom_bar(stat="identity", position="dodge") +
  facet_wrap(~Area) +
  scale_fill_manual(values=c("Minor"="green", "Severe"="red", "Fatal"="black")) +
  theme_minimal() +
  labs(
    title="Helmet Use vs Severe Injury by Area",
    x="Helmet Use",
    y="Percentage of Accidents",
    fill="Severity"
  )

p_helmet
```

Shiny App Application Implementation

The RTA Shiny app was developed to provide interactive visualizations of accident severity, allowing users to explore patterns by Area, Vehicle type, Gender, Helmet use, and Driving experience. This section describes the implementation step by step.

App Overview

The Shiny app provides:

- **Dynamic filtering:** Users can select Area, Gender, or Helmet use to focus on specific groups.
- **Variable selection:** Users can choose **X-axis** and **Y-axis** variables to explore relationships.
- **Interactive plots:** Visualizations are **interactive**, with hover tooltips showing exact values and percentages.
- **Multi-factor analysis:** Combined filters allow exploration of **high-risk combinations**, e.g., rural male motorcyclists without helmets.

Shiny UI Implementation

The **User Interface (UI)** was designed using fluidPage() for a clean layout:

```
library(shiny)
library(plotly)

ui <- fluidPage(
  titlePanel("RTA Accident Analysis Shiny App"),

  sidebarLayout(
    sidebarPanel(
      # Filters for multi-factor analysis
      selectInput("filter_area", "Select Area:", choices=c("All", levels(df_clean$Area))),
      selectInput("filter_gender", "Select Gender:", choices=c("All",
levels(df_clean$Gender))),
```

```
    selectInput("filter_helmet", "Select Helmet Use:", choices=c("All",
levels(df_clean$Helmet_use))),

    # Dynamic variable selection for plotting
    selectInput("x_var", "Select X-axis Variable:",

choices=c("Area","Vehicle_type","Gender","Helmet_use","Driving_experience")),
    selectInput("y_var", "Select Y-axis Variable:",
                choices=c("Severity","Count"))
  ),

  mainPanel(
    plotlyOutput("dynamic_plot")  # Displays the interactive plot
  )
 )
)
```

Explanation:

- selectInput() allows users to **choose filters or variables** for analysis.
- plotlyOutput() renders the plot interactively.
- sidebarLayout() keeps controls on the side, with the plot in the main panel.

Shiny Server Implementation

The **Server Logic** handles **filtering, aggregation, and plotting** dynamically:

```
server <- function(input, output) {

 # Reactive filtered data based on user selection
 filtered_data <- reactive({
  data <- df_clean

  if(input$filter_area != "All") { data <- data %>% filter(Area == input$filter_area) }
  if(input$filter_gender != "All") { data <- data %>% filter(Gender ==
input$filter_gender) }
  if(input$filter_helmet != "All") { data <- data %>% filter(Helmet_use ==
input$filter_helmet) }

  data
 })

 # Render dynamic plot
 output$dynamic_plot <- renderPlotly({

  plot_data <- filtered_data() %>%
   group_by_at(vars(input$x_var, input$y_var)) %>%
   summarise(Count = n()) %>%
   ungroup() %>%
   mutate(Percent = Count / sum(Count) * 100)

  p <- ggplot(plot_data, aes_string(x=input$x_var, y="Percent", fill=input$y_var)) +
```

```
    geom_bar(stat="identity", position="dodge") +
    scale_fill_manual(values=c("Minor"="green","Severe"="red","Fatal"="black")) +
    theme_minimal() +
    labs(y="Percentage of Accidents", fill=input$y_var)

  ggplotly(p, tooltip=c("x","y","fill"))
 })
}
```

Explanation:

- reactive() ensures that **data updates automatically** whenever filters change.
- group_by_at(vars(...)) allows **dynamic grouping** based on user selection.
- geom_bar(stat="identity") plots percentages rather than raw counts.
- ggplotly() converts the ggplot into an **interactive Plotly plot**, showing hover tooltips.

Running the Shiny App

```
# Launch the app
shinyApp(ui, server)
```

- This command **starts the Shiny app locally**.
- Users can explore **interactive charts** and **apply filters** to analyze accident patterns.

Deployment

The app was deployed to **ShinyApps.io**, a free hosting platform for Shiny apps:

1. Save the app as app.R in a project folder.
2. Install and load the rsconnect package:

```
install.packages("rsconnect")
library(rsconnect)
```

3. Connect your ShinyApps.io account:

```
rsconnect::setAccountInfo(name='sabashahbaz',
                          token='YOUR_TOKEN',
                          secret='YOUR_SECRET')
```

4. Deploy the app:

```
rsconnect::deployApp('path/to/your/app/folder')
```

- Once deployed, the app is accessible online: https://sabashahbaz.shinyapps.io/rta_shiny_app/

GitHub Repository

To make your RTA Shiny App project **publicly accessible and well-organized**, a GitHub repository was created. This allows others to **view, reuse, and collaborate** on your code and data.

Repository Overview

Repository Name: RTA_Shiny_App

Purpose:

- Store all project files, including **dataset, R scripts, and Shiny app code**.
- Track versions of the code as updates are made.
- Provide a link to the **live Shiny app** for easy access.

GitHub Link: https://github.com/sabashahbaz/RTA_Shiny_App

Conclusion

The RTA Shiny App project successfully transforms raw road traffic accident data into **actionable insights** through interactive and informative visualizations. By combining **data preparation, percentage and uncertainty calculations, dynamic plots, and interactive filters**, the project highlights **high-risk areas, time patterns, vehicle types, and behavioral factors** contributing to severe injuries.

Key outcomes include:

- **Identification of high-risk zones:** Rural areas and busy market roads show higher proportions of severe accidents, emphasizing the need for targeted infrastructure improvements and safety campaigns.
- **Temporal trends:** Peaks in severe injuries on specific days, such as weekends, suggest the importance of timing interventions and enforcement.
- **Vehicle and behavioral insights:** Motorcycles and heavy vehicles are frequently associated with severe injuries, and helmet use significantly reduces risk.
- **Interactive exploration:** The Shiny App allows users to dynamically filter and analyze multiple factors, making the insights **accessible to policymakers, health professionals, and the general public**.

The combination of **static graphs, interactive plots, and GitHub documentation** ensures that the project is **transparent, reproducible, and shareable**. By presenting data in a clear, visual, and interactive format, this project provides a **practical tool for road safety awareness, prevention strategies, and evidence-based policy planning**.

Overall, the RTA Shiny App demonstrates how **data-driven insights can guide real-world interventions** to reduce the severity and frequency of road traffic accidents, improving safety outcomes in both urban and rural settings.