**QUESTION:**

Use the ‘shiny’ module in R to create an interactive dashboard for the dataset assigned to you.

**CODE:**

library(shiny)

library(tidyverse)

# Load the data

df <- read.csv("VehicleFailureData.csv")

# Remove commas and convert Mileage\_at\_Failure to numeric

df$Mileage\_at\_Failure <- as.numeric(gsub(",", "", df$Mileage\_at\_Failure))

# Define UI

ui <- fluidPage(

  titlePanel("Vehicle Failure Data"),

  sidebarLayout(

    sidebarPanel(

      selectInput(inputId = "state",

                  label = "Select State:",

                  choices = unique(df$State),

                  selected = "CA"),

      sliderInput(inputId = "mileage",

                  label = "Mileage Range:",

                  min = 0,

                  max = 50000,

                  value = c(10000, 30000)),

      checkboxInput(inputId = "failed",

                    label = "Show Failed Vehicles Only",

                    value = FALSE),

      hr(),

      helpText("Created by Afraaz Hussain (Admission Number: 20BDS0374)")

    ),

    mainPanel(

      plotOutput(outputId = "scatterplot"),

      tableOutput(outputId = "table")

    )

  )

)

# Define server

server <- function(input, output) {

  # Filter data based on inputs

  filteredData <- reactive({

    df %>%

      filter(State == input$state,

             Mileage\_at\_Failure >= input$mileage[1],

             Mileage\_at\_Failure <= input$mileage[2]) %>%

      if(input$failed) filter(Failure\_Month > 0) else .

  })

  # Scatter plot

  output$scatterplot <- renderPlot({

    ggplot(filteredData(), aes(x = Mileage\_at\_Failure, y = Labor\_Cost)) +

      geom\_point()

  })

  # Table

  output$table <- renderTable({

    filteredData() %>%

      select(Vehicle\_Number, State, Failure\_Month, Mileage\_at\_Failure, Labor\_Cost) %>%

      head(10)

  })

}

# Run the app

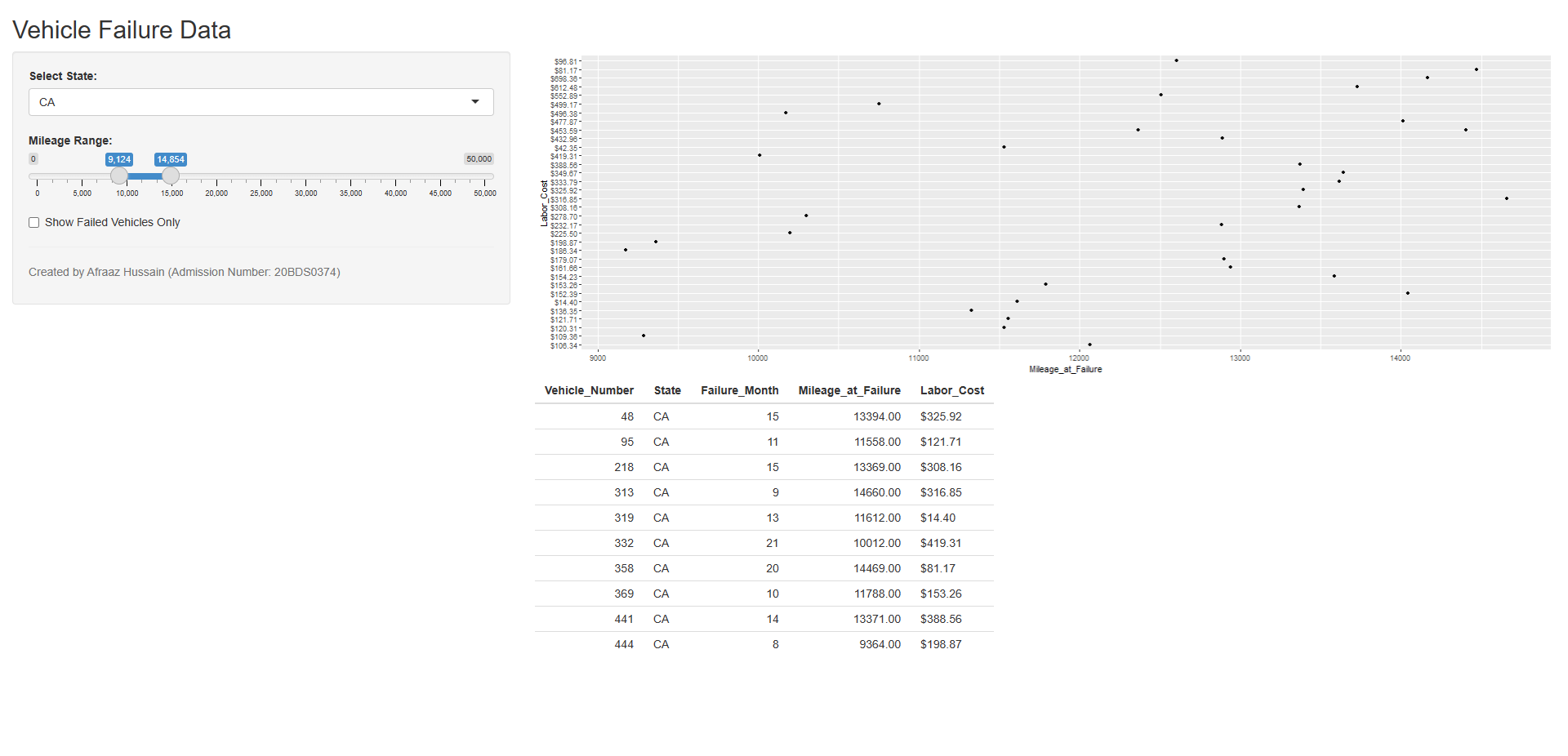
shinyApp(ui = ui, server = server)

**DESCRIPTION:**

* Numeric Input: *min\_mileage* and *max\_mileage* - These filters allow the user to specify a minimum and maximum mileage for filtering the data. The app will only show records that have a mileage value within this range.
* Numeric Input: *min\_labor\_hours* and *max\_labor\_hours* - These filters allow the user to specify a minimum and maximum labor hours for filtering the data. The app will only show records that have a labor hours value within this range.
* Numeric Input: *min\_labor\_cost* and *max\_labor\_cost* - These filters allow the user to specify a minimum and maximum labor cost for filtering the data. The app will only show records that have a labor cost value within this range.
* Numeric Input: *min\_material\_cost* and *max\_material\_cost* - These filters allow the user to specify a minimum and maximum material cost for filtering the data. The app will only show records that have a material cost value within this range.
* Select Input: state - This filter allows the user to select a specific state to filter the data by. The app will only show records that belong to the selected state.
* Checkbox Input: *show\_summary\_stats* - This widget allows the user to toggle the display of summary statistics on or off. If checked, the app will show summary statistics for the current filtered dataset.
* DataTable Output: *vehicle\_table* - This output displays the current filtered dataset in a tabular format, with columns for each of the variables in the dataset.
* VerbatimTextOutput: *summary\_stats* - This output displays the summary statistics for the current filtered dataset, if the *show\_summary\_stats* checkbox is checked.

**OUTPUT:**

* All the vehicles in CA, with a mileage from 9,124 to 14, 854:



* All the vehicles in FL, with a mileage from 14,690 to 25,182:

Graphical user interface, application, table

Description automatically generated

**LINK TO THE DASHBOARD:**