

DSMR MINI PROJECT: STATISTICAL CALCULATOR

INDEX

1. Aim.....	2
2. Students and their contribution.....	3
3. Tool used in the project.....	4
4. Execution Code in R.....	5
5. Resulted App.....	8
6. Data testing.....	9
7. Visualization testing.....	12
8. Statistical testing.....	13
9. URL for the app.....	14
10. Future Scope.....	15
11. Conclusion.....	16

AIM:

Create a Shiny Web App to calculate mean, median, mode and standard deviation from the file uploaded by the user. Visualize the parameters chosen by the user from the dropdown menu. Plot a normal plot, Scatter plot, Box plot and Histogram.

STUDENTS AND THEIR CONTRIBUTION

AKSHAT SHAH 19162121036

- File input from user

ASHIL SHAH 19162121037

- UI Code

DISHWA SHAH 19162121038

- UI and Server Code
- Testing

TOOL USED IN THE PROJECT: RSTUDIO AND SHINY WEB APP

RStudio:

- RStudio is an Integrated Development Environment (IDE) for R, a programming language for statistical computing and graphics.
- It is available in two formats: RStudio Desktop is a regular desktop application while RStudio Server runs on a remote server and allows accessing RStudio using a web browser.

Shiny Web App:

- Shiny is an R package that makes it easy to build interactive web apps straight from R.
- You can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards.
- You can also extend your Shiny apps with CSS themes, html widgets, and JavaScript actions.
- Shiny combines the computational power of R with the interactivity of the modern web.
- Shiny apps are easy to write. No web development skills are required.

EXECUTION CODE IN R

```

library(DT)
library(shiny)
library(shinythemes)
library(summarytools)
library(ggplot2)
library(GGally)

ui <- fluidPage(theme = shinytheme("flatly"),
  tags$style('.container-fluid {
    background-color: #d2d9d7
  }'),
  titlePanel(
    h2("Statistical Calculator", align = "center")
  ),
  br(),
  br(),
  sidebarPanel(" ",
    fileInput(
      "dataset",
      "Choose a CSV file",
      multiple = FALSE,
      accept=c('text/csv', 'text/comma-
separated-values,text/plain', '.csv'),
      width = NULL,
      buttonLabel = "Browse",
      placeholder = "No file selected"
    ),
    checkboxInput("header", "Header", TRUE),
    uiOutput("columns")),

  mainPanel(
    tabsetPanel(
      tabPanel("Data",
        h4("Data Table", align = "center"),
        dataTableOutput("data")),

      tabPanel("Summary",
        h4("Structure", align = "center"),
        br(),
        verbatimTextOutput("stc"),
        br(),
        h4("Summary", align = "center"),
        br(),
        verbatimTextOutput("summary")),

      tabPanel("Descriptive Statistics",

```

```

align = "center"),
    h4("Descriptive Statistics of Data",
    br(),
    verbatimTextOutput("des")),

    tabPanel("Visualizations",
      h4("Histogram", align = "center"),
      br(),
      plotOutput("hist"),
      h4("Box plot", align = "center"),
      br(),
      plotOutput("box"),
      h4("Scatter Plot", align = "center"),
      br(),
      plotOutput("scatter"),
      h4("QQ-Plot", align = "center"),
      br(),
      plotOutput("qq")),

    tabPanel("Statistics",
      h4("Mean", align = "center"),
      br(),
      verbatimTextOutput("mean"),
      br(),
      h4("Median", align = "center"),
      verbatimTextOutput("median"),
      br(),
      h4("Mode", align = "center"),
      verbatimTextOutput("mode"),
      br(),
      h4("Standard deviation", align =
"center"),
      verbatimTextOutput("sd")))))

server <- function(input, output, session) {
  df <- reactive(read.csv(input$dataset$datapath, header =
input$header))
  data <- reactive(na.omit(df()))
  mod <- function(x){which.max(tabulate(x))}
  output$data <- renderDataTable(
    data(),
    server = TRUE
  )
  output$stc <- renderPrint({
    str(data())
  })
  output$columns <- renderUI({
    selectInput("columns", "Choose a column",
      choices <- colnames(data()),

```

```

        multiple = F)
    })
    output$summary <- renderPrint({
      dfSummary(data())
    })
    output$des <- renderPrint({
      descr(data())
    })

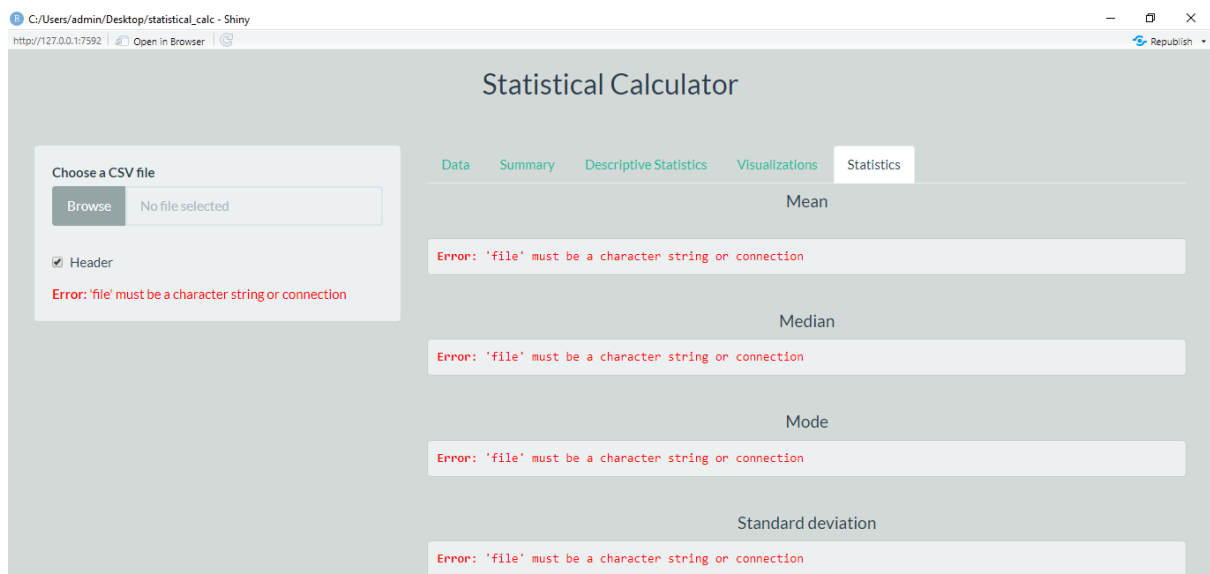
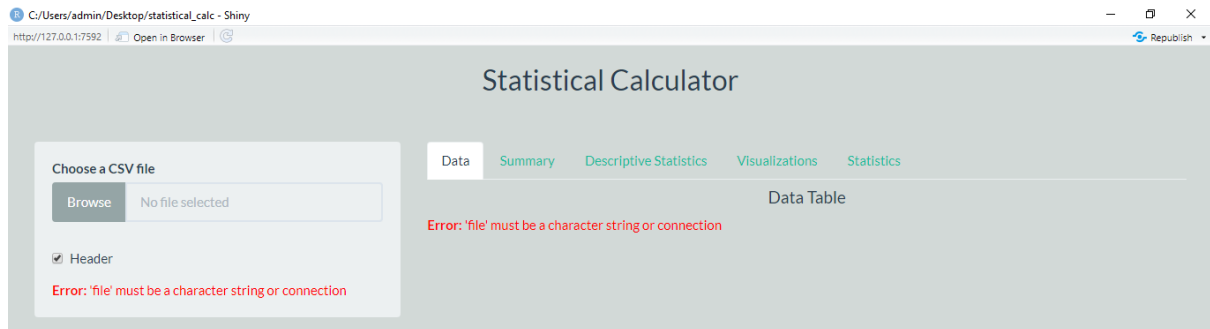
    output$mean <- renderPrint({
      mean(data()[,input$columns])
    })
    output$median <- renderPrint({
      median(data()[,input$columns])
    })
    output$mode <- renderPrint({
      mod(data()[,input$columns])
    })
    output$sd <- renderPrint({
      sd(data()[,input$columns])
    })

    output$hist <- renderPlot({
      hist(data()[,input$columns])
    })
    output$box <- renderPlot({
      boxplot(data()[,input$columns])
    })
    output$scatter <- renderPlot({
      plot(data()[,input$columns])
    })
    output$qq <- renderPlot({
      qqnorm(data()[,input$columns])
      qqline(data()[,input$columns])
    })
  }

  shinyApp(ui = ui, server = server)

```

RESULTED APP



DATA TESTING

Upload a CSV file.

You can choose the number of entries you want to display at once.

You can jump to a particular page through the panel in the bottom.

The screenshot shows the 'Statistical Calculator' application interface. On the left, there is a sidebar with options to 'Choose a CSV file' (Browse, Upload complete), a checkbox for 'Header' (checked), and a 'Choose a column' dropdown (set to 'Car'). The main area displays the 'Data Table' with columns: Car, MPG, Cylinders, Displacement, Horsepower, Weight, Acceleration, and Model. The table shows 10 entries, with the first row being the header. The bottom of the table has pagination controls showing 'Showing 1 to 10 of 406 entries' and page numbers 1 through 41.

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model
1	Chevrolet Chevelle Malibu	18	8	307	130	3504	12	70
2	Buick Skylark 320	15	8	350	165	3693	11.5	70
3	Plymouth Satellite	18	8	318	150	3436	11	70
4	AMC Rebel SST	16	8	304	150	3433	12	70
5	Ford Torino	17	8	302	140	3449	10.5	70
6	Ford Galaxie 500	15	8	429	198	4341	10	70
7	Chevrolet Impala	14	8	454	220	4354	9	70
8	Plymouth Fury iii	14	8	440	215	4312	8.5	70
9	Pontiac Catalina	14	8	455	225	4425	10	70
10	AMC Ambassador DPL	15	8	390	190	3850	8.5	70

You can uncheck the 'Header' if you do not want to display the first row as a header.

The screenshot shows the 'Statistical Calculator' application interface. On the left, the 'Header' checkbox is unchecked. The 'Choose a column' dropdown is set to 'V1'. The main area displays the 'Data Table' with columns: V1, V2, V3, V4, V5, V6, V7, and V8. The table shows 10 entries, with the first row being data. The bottom of the table has pagination controls showing 'Showing 1 to 10 of 406 entries' and page numbers 1 through 41.

	V1	V2	V3	V4	V5	V6	V7	V8
1	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model
2	Chevrolet Chevelle Malibu	18	8	307	130	3504	12	70
3	Buick Skylark 320	15	8	350	165	3693	11.5	70

You can type any number or word in 'Search' to find the similar ones in the whole dataset.

Statistical Calculator

Choose a CSV file
Browse cars.csv Upload complete

☒ Header

Choose a column
Car

Data Table

Show 10 entries Search: buick

	Car	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Model
2	Buick Skylark 320	15	8	350	165	3693	11.5	70
20	Buick Estate Wagon (sw)	14	8	455	225	3086	10	70
76	Buick LeSabre Custom	13	8	350	155	4502	13.5	72
93	Buick Century 350	13	8	350	175	4100	13	73

You can view the structure and the summary of the entire dataset under 'Summary'.

Statistical Calculator

Choose a CSV file
Browse cars.csv Upload complete

☒ Header

Choose a column
Car

Structure

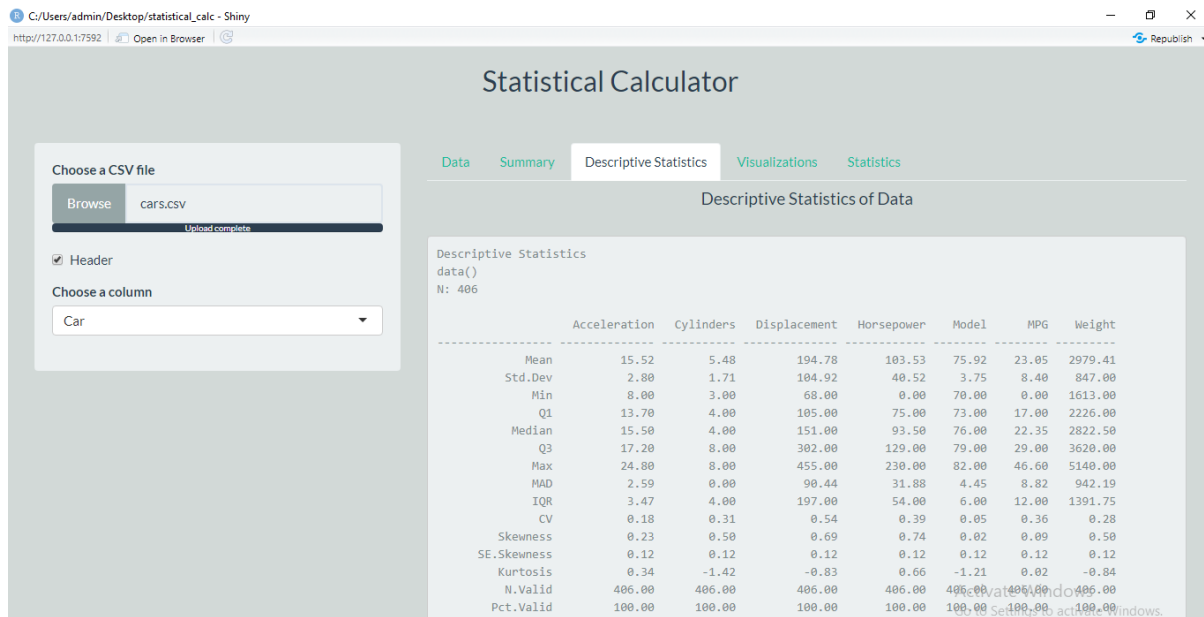
```
'data.frame': 406 obs. of 8 variables:
 $ Car      : chr  "Chevrolet Chevelle Malibu" "Buick Skylark 320" "Plymouth Satellite" "AMC Rebel SST" ...
 $ MPG      : num  18 15 18 16 17 15 14 14 15 ...
 $ Cylinders : int   8  8  8  8  8  8  8  8  8 ...
 $ Displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ Horsepower : int  130 165 150 150 140 198 220 215 225 190 ...
 $ Weight     : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
 $ Acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ Model      : int  70 70 70 70 70 70 70 70 70 ...
```

Summary

Data Frame Summary
data()
Dimensions: 406 x 8
Duplicates: 0

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	Car [character]	1. Toyota Corolla 2. Ford Pinto 3. AMC Matador 4. Ford Maverick 5. Volkswagen Rabbit 6. AMC Gremlin 7. AMC Hornet 8. Chevrolet Chevette 9. Chevrolet Impala 10. Peugeot 504 [298 others]	9 (2.2%) 6 (1.5%) 5 (1.2%) 5 (1.2%) 5 (1.2%) 4 (1.0%) 4 (1.0%) 4 (1.0%) 4 (1.0%) 4 (1.0%) 356 (87.7%)		406 (100.0%)	0 (0.0%)
2	MPG [numeric]	Mean (sd) : 23.1 (8.4) min < med < max: 0 < 22.4 < 46.6 IQR (CV) : 12 (0.4)	130 distinct values		406 (100.0%)	0 (0.0%)

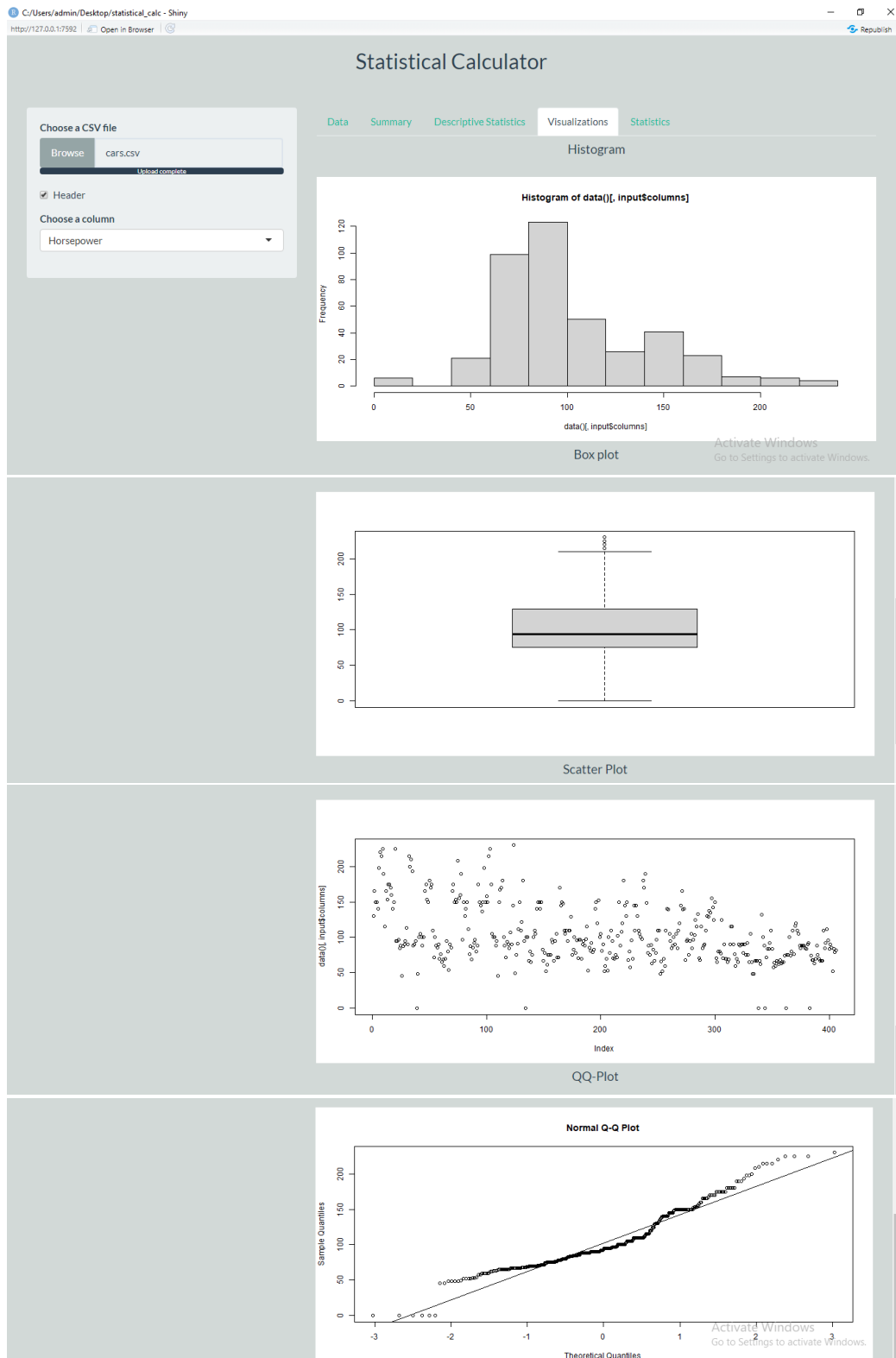
You can also view the Descriptive Statistics of the entire dataset.



VISUALIZATION TESTING

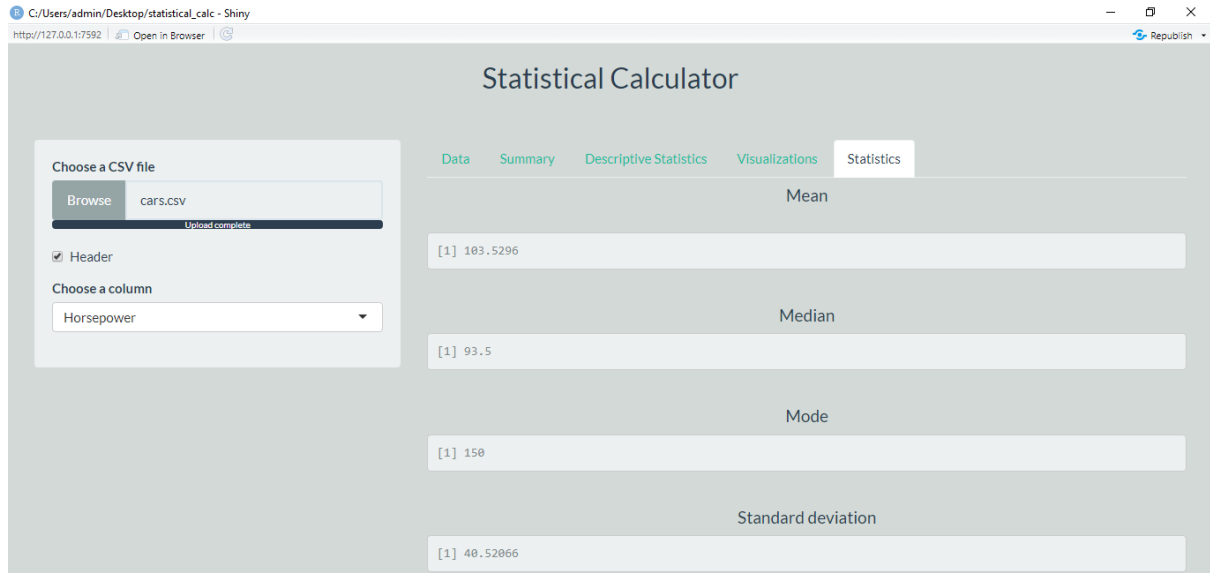
Select a column from the dataset whose statistics you want to calculate.

You can then view the Histogram, Box Plot, scatter Plot and QQNorm Plot of the selected column/parameter.



STATISTICAL TESTING

You can view the calculated mean, median, mode and standard deviation of the parameter you selected under 'Statistics'.



URL FOR THE APP

https://dishwa.shinyapps.io/statistical_calc/

FUTURE SCOPE

- More statistics can be calculated.
- More plots can be displayed.
- Different file formats can be supported.
- Statistics of multiple rows can be calculated at once.

CONCLUSION

We created an in-app calculator that displays mean, median, mode and standard deviation of the parameter selected by the user. The app also displays various plots to help understand the data of the user-uploaded file better.