

Mini Project 2

PSTAT100: Data Science Concepts and Analysis

Instructor: Ali Abuzaid

2024-11-15

STUDENT NAME


- Bahaar Ahuja (bahaar)
- Veronica Stremper (vstremper)
- Sophie Lian (sophielian)
- Paridhi Jay Singh (paridhijsingh)

Instructions

- This mini project aims to familiarize you with real-life data sourced from various resources.
- The mini project includes narrative questions. While these questions are primarily based on lecture material and prerequisites, they may also require independent thinking and investigation.
- Collaborate in groups of **3-4** students from the **same discussion session**; individual submissions **will not be accepted**.
- Ensure that all R code, mathematical formulas, and workings are presented clearly and appropriately.
- Please submit a .zip file of the folder you create, containing your **app.R** file along with the report.

Information on Grading To grade your project submissions, the grader will download your project files and run the app you created locally.

Important: If your app fails to open due to issues like incorrect file format or improper zipping, your group will receive a score of ZERO for this project. It's essential to submit a functioning app for fair grading. Thank you for your understanding!**

 Due Date

Due Date: Friday, November 15, 2024, 11:59 PM

1 Overview

Project Title: Interactive Exploration of the ‘Iris’ Dataset

1.1 Objectives:

Data Exploration: Students will gain hands-on experience in data exploration and visualization using a classic dataset available in R.

Shiny App Development: Students will learn the basics of building a Shiny app, including UI design and server logic.

User Interaction: Students will implement interactive features allowing users to filter and visualize the dataset.

Presentation of Results: Students will summarize their findings and demonstrate the app to their peers. Dataset:

1.2 Dataset

Choose any dataset available in R (e.g., `mtcars`, `iris`, `diamonds`, etc.). You can load datasets directly using built-in R functions.

2 Tasks:

2.1 Data Loading and Preparation:

1- Load your chosen dataset using the following R code:

```
1 data()
2 data(iris)
3 head(iris)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

```
1 summary(iris)
```

```
      Sepal.Length    Sepal.Width    Petal.Length    Petal.Width
Min.   :4.300      Min.   :2.000      Min.   :1.000      Min.   :0.100
1st Qu.:5.100      1st Qu.:2.800      1st Qu.:1.600      1st Qu.:0.300
Median :5.800      Median :3.000      Median :4.350      Median :1.300
Mean   :5.843      Mean   :3.057      Mean   :3.758      Mean   :1.199
3rd Qu.:6.400      3rd Qu.:3.300      3rd Qu.:5.100      3rd Qu.:1.800
Max.   :7.900      Max.   :4.400      Max.   :6.900      Max.   :2.500

      Species
setosa   :50
versicolor:50
virginica :50
```

```
1 str(iris)
```

```
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

2- Explore the dataset to understand its structure and perform any necessary data cleaning. Use functions like `summary(mtcars)` and `str(mtcars)` for insights.

2.2 Shiny App Development

User Interface (UI):

1- Create a sidebar layout that allows users to select:

- A variable to analyze.
- Type of plot (e.g., scatter plot, histogram, box plot).
- Display the selected plot and relevant summary statistics.

```

1 library(shiny)
2 library(ggplot2)
3 library(dplyr)
4
5 # Define UI
6 ui <- fluidPage(
7
8   titlePanel("Iris Dataset Analysis"),
9
10  sidebarLayout(
11
12    sidebarPanel(
13      selectInput("variable", "Select a variable to analyze:",
14                  choices = names(iris)[1:4]),
15
16      selectInput("plot_type", "Select plot type:",
17                  choices = c("Histogram", "Box Plot", "Scatter Plot")),
18
19      checkboxGroupInput("species", "Select Species:",
20                         choices = levels(iris$Species),
21                         selected = levels(iris$Species)),
22
23      sliderInput("value_range", "Select range of values:",
24                  min = min(iris$Sepal.Length), max = max(iris$Sepal.Length),
25                  value = range(iris$Sepal.Length))
26    ),
27
28    mainPanel(
29      plotOutput("plot"),
30      verbatimTextOutput("summary")
31    )
32  )
33 )

```

Server Logic: Write server code to respond to user inputs and generate the selected plot dynamically.

```

1 server <- function(input, output, session) {
2
3
4   observe({
5     req(input$variable)

```

```

6   var <- input$variable
7   updateSliderInput(session, "value_range",
8                     min = min(iris[[var]], na.rm = TRUE),
9                     max = max(iris[[var]], na.rm = TRUE),
10                    value = range(iris[[var]], na.rm = TRUE))
11 })
12
13 filtered_data <- reactive({
14   req(input$variable, input$value_range, input$species)
15   iris %>%
16     filter(Species %in% input$species) %>%
17     filter(get(input$variable) >= input$value_range[1],
18            get(input$variable) <= input$value_range[2])
19 })
20
21 output$plot <- renderPlot({
22   data <- filtered_data()
23   req(input$plot_type)
24
25   if (input$plot_type == "Histogram") {
26     ggplot(data, aes_string(x = input$variable)) +
27       geom_histogram(binwidth = 0.3, fill = "blue", color = "black") +
28       labs(title = paste("Histogram of", input$variable),
29            x = input$variable, y = "Count")
30
31   } else if (input$plot_type == "Box Plot") {
32     ggplot(data, aes_string(x = "Species", y = input$variable)) +
33       geom_boxplot(fill = "lightgreen") +
34       labs(title = paste("Box Plot of", input$variable, "by Species"),
35            x = "Species", y = input$variable)
36
37   } else if (input$plot_type == "Scatter Plot") {
38     ggplot(data, aes_string(x = input$variable,
39                             y = "Petal.Length", color = "Species")) +
40       geom_point() +
41       labs(title = paste("Scatter Plot of", input$variable, "vs Petal Length"),
42            x = input$variable, y = "Petal Length")
43   }
44 })
45
46 output$summary <- renderPrint({
47   req(filtered_data())

```

```

48     summary(filtered_data()[[input$variable]])
49   })
50 }
51
52 shinyApp(ui = ui, server = server)

```

2.3 Interactivity

- 1- Implement features such as sliders to adjust numerical variables and checkboxes for selection.
- 2- Consider adding tooltips or hover functionality for enhanced user experience.

3 Documentation and Presentation

Prepare a brief report summarizing the project, including:

- 1- Objectives of the app.

The primary objective of this app is to provide users with a platform to explore and analyze the Iris dataset, which includes measurements for different flower features of three iris species: Setosa, Versicolor, and Virginica. The app allows users to: Visualize relationships and distributions of various flower features. Compare attributes across different iris species. Gain insights into the statistical properties of each attribute through dynamic visualizations and summary statistics.

- 2- Description of the functionality.

Our app offers a user-friendly interface for visualizing and exploring the Iris dataset. Through this app, users can select specific variables, three different plots, and varied data ranges to explore the dataset dynamically, allowing for deeper insights. Through the dropdown menu, the app offers variable and plot-type selection, through which users can choose a specific attribute from the dataset-such as Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width-to focus their analysis.

We can also access three different plot types, such as the boxplot, histogram, and scatterplots through the dropdown menu. The Histogram displays the frequency distribution of any variable of the four (Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width), offering insights on its spread and frequency. The Boxplot enables a comparison of the chosen variable across the three species of the Iris Plant, (Setosa, Virginica, and Versicolor), illustrating the differences in the distribution. The Scatterplot represents a selected variable against Petal.Length, which points color-coded by species, allowing users to explore potential correlations within the dataset.

We have also implemented checkboxes for Setosa, Virginica, and Versicolor which will allow users to focus on specific species using the species selection feature. Users can choose to view data for one, two or all species at once, which will make it easy to compare and contrast the species-specific pattern. To make the data more readable, we added a value range slider, which users could adjust based on minimum and maximum values of the selected variable, to narrow the display to specific ranges. This range-based filtering helps focus the analysis on specific subsets of data, such as only the flowers within a particular sepal length range.

Finally, our app also includes a summary statistics section at the end of the plots, which displays key measures like-minimum value, maximum value, mean, median, and quartiles-for the selected variable. As users adjust the sliders and species selections, these statistics refresh to provide an updated summary of the filtered data. In a Nutshell, All selections update in real-time, providing an interactive plot display that reflects changes to the selected variable, plot type, species, or value range.

3- Key findings or insights derived from the data.

Some insights that can be derived from the data is when analyzing the sepal and petal lengths and widths Setosa tends to have the smallest sepal and petal measurements. Whereas Virginica tends to have the highest measurements for petal length and width. Additionally, using different plots such as a boxplot it can be seen that the Setosa species has the smallest median value for sepal length, petal length, and petal width, but the largest median value for petal length. We can also use this application to analyze the relationship between sepal length, sepal width, and petal width against the variable of petal length of different iris species. For the Virginica and Versicolor species, there is a strong positive correlation between sepal length and petal length, meaning that flowers of these species tend to have longer petals as their sepals increase in length. While on the other hand, the Setosa species has almost no correlation between sepal length and petal length.

4 References

- 1- Shiny Documentation: [Welcome to Shiny](#)
- 2- Hadley Wickham (2021) Mastering Shiny, O'Reilly Media. (<https://mastering-shiny.org/>)