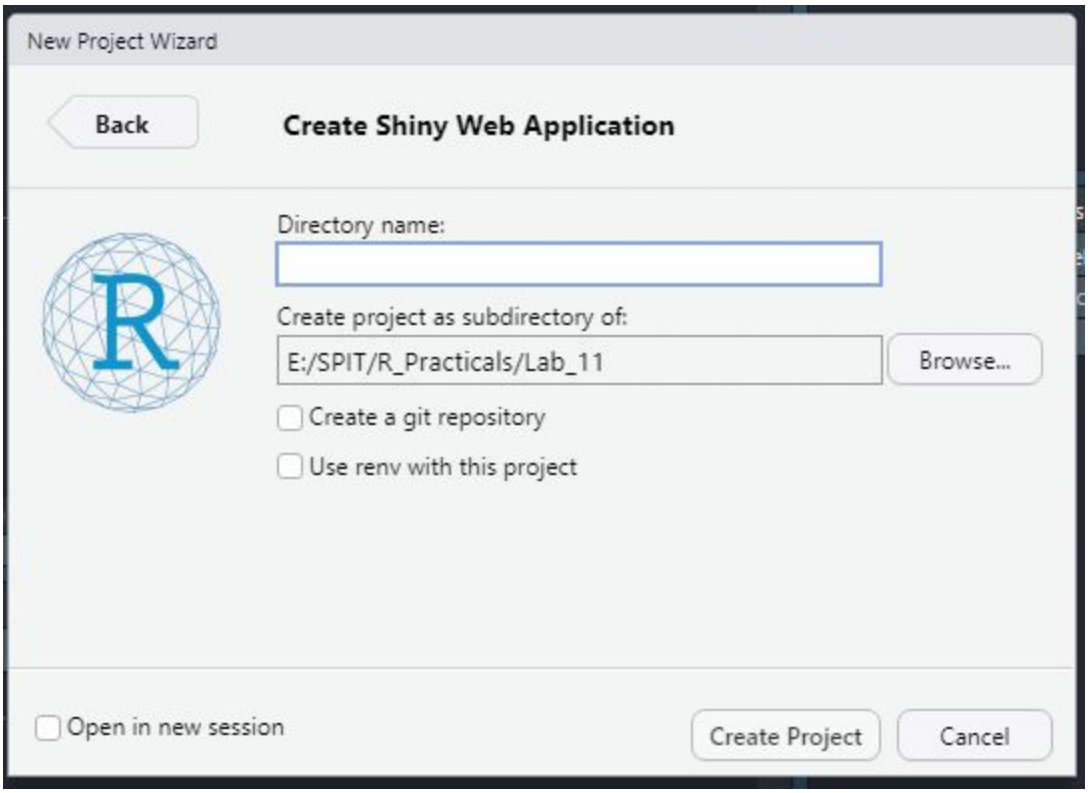
**NAME: Hammad Ansari** **ROLL NO: 2018450002**

# Experiment No. 11

**Title**: Shiny R Applications

**Problem**:

1. Drawing histograms for iris dataset in R using Shiny- By varying the input like selecting a column of iris dataset and no of beans, Histogram should get created as per the user inputs.
    a. Create a new project in R Studio and Select type as Shiny web application.



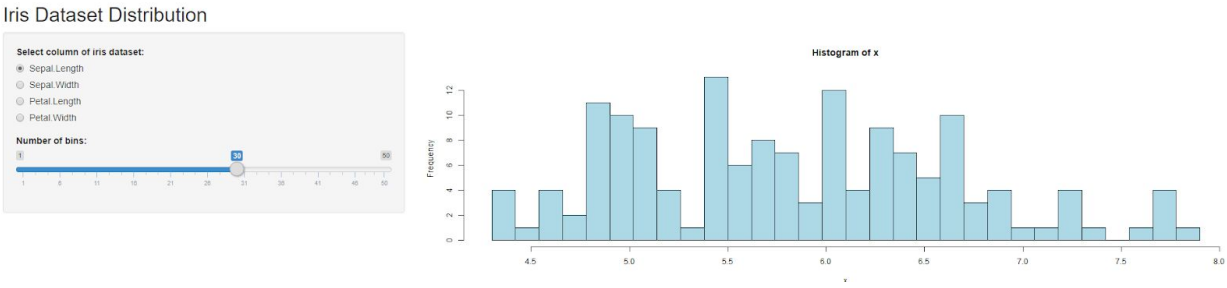

    b. It creates two scripts in R Studio named ui.R and server.R.

```
#

# This is a Shiny web application. You can run the application by clicking

# the 'Run App' button above.
```

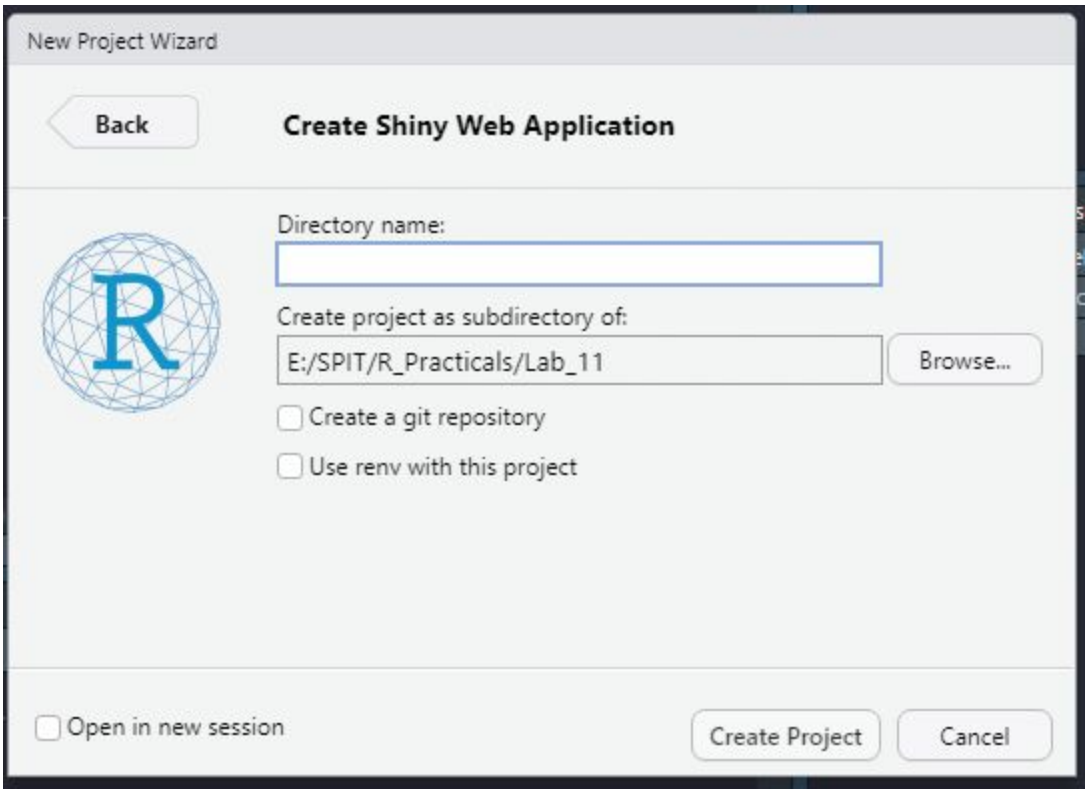
```
#

# Find out more about building applications with Shiny here:

#

#    http://shiny.rstudio.com/

#


library(shiny)


# Define UI for an application that draws a histogram

ui <- fluidPage(


# Application title

   titlePanel("Iris Dataset Distribution"),


# Sidebar with a slider input for number of bins

   sidebarLayout(

      sidebarPanel(

         radioButtons("p", "Select column of iris dataset:",

           list("Sepal.Length" = 'a', "Sepal.Width" = 'b', "Petal.Length" = 'c',
"Petal.Width" = 'd')),

           sliderInput("bins", "Number of bins:", min = 1, max = 50, value =
30)),
```

```
# Show a plot of the generated distribution
        mainPanel(
          plotOutput("distPlot")
        )
    )
)


# Define server logic required to draw a histogram
server <- function(input, output) {
  output$distPlot <- renderPlot({
    if (input$p == 'a') { i <- 1 }
    if (input$p == 'b') { i <- 2 }
    if (input$p == 'c') { i <- 3 }
    if (input$p == 'd') { i <- 4 }
    x <- iris[, i]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)
    hist(x, breaks = bins, col = 'darkgray', border = 'white')
  })
}


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

c. Output




2. Drawing Scatterplots for iris dataset in R using Shiny - Allow users to select any feature from the iris dataset on the x-axis and y-axis and scatterplot should be created as per user input.
   a. Create a new project in R Studio and Select type as Shiny web application.




   b. It creates two scripts in R Studio named ui.R and server.R.

#

```
# This is a Shiny web application. You can run the application by clicking

# the 'Run App' button above.

#

# Find out more about building applications with Shiny here:

#

#    http://shiny.rstudio.com/

#


library(shiny)


# Define UI for application that draws a histogram

ui <- fluidPage(


# Application title

   titlePanel("Iris Scatter Plot"),


# Sidebar with a slider input for number of bins

   sidebarLayout(

      sidebarPanel(

         radioButtons("x", "Select x-axis:",

                      list("Sepal.Length" = 'a', "Sepal.Width" = 'b', "Petal.Length"
= 'c', "Petal.Width" = 'd')),

         radioButtons("y", "Select y-axis:",
```

```
                        list("Sepal.Length" = 'e', "Sepal.Width" = 'f', "Petal.Length"
= 'g', "Petal.Width" = 'h'))

          ),


# Show a plot of the generated distribution

       mainPanel(

         plotOutput("distPlot")

       )

   )

)


# Define server logic required to draw a histogram

server <- function(input, output) {


  output$distPlot <- renderPlot({

    if (input$x == 'a') { i <- 1 }

    if (input$x == 'b') { i <- 2 }

    if (input$x == 'c') { i <- 3 }

    if (input$x == 'd') { i <- 4 }

    if (input$y == 'e') { j <- 1 }

    if (input$y == 'f') { j <- 2 }

    if (input$y == 'g') { j <- 3 }

    if (input$y == 'h') { j <- 4 }
```

```
    x_axis <- iris[, i]
    y_axis <- iris[, j]
    plot(x_axis, y_axis, col="blue")
  })
}

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

c. Output