Problem 3

Yufei Yin

ui.R

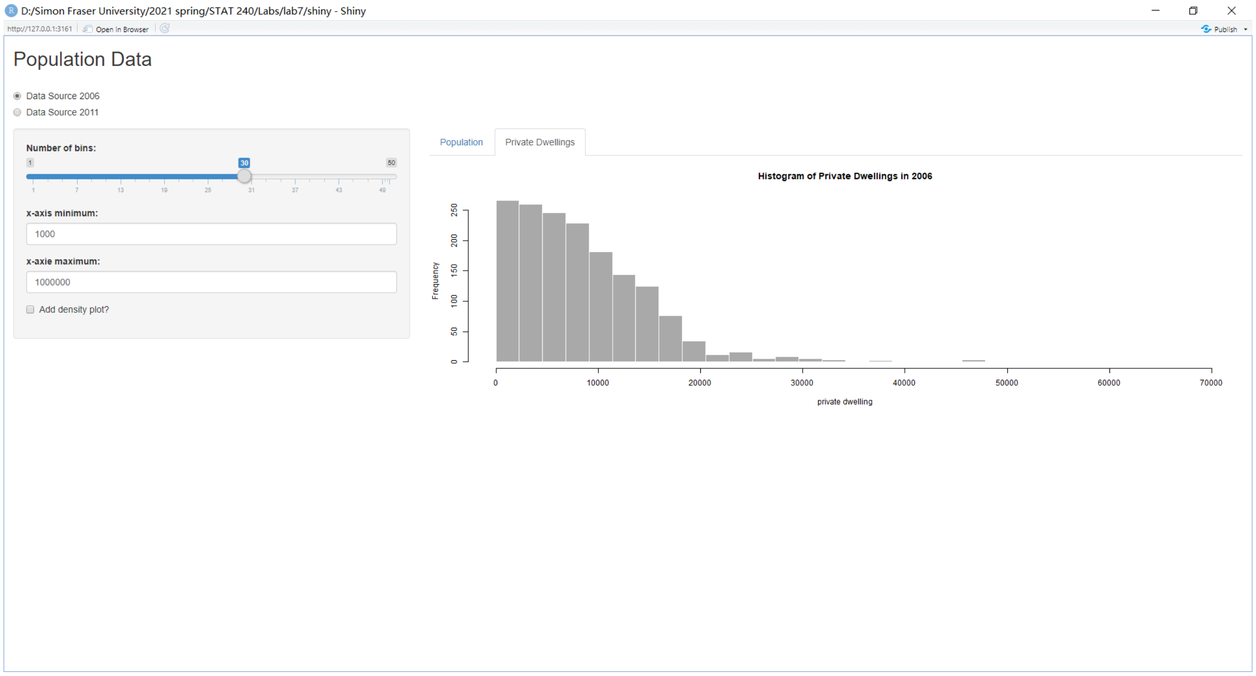
```r
library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Population Data"),
  # Include radio buttons  (Only one of these can be selected)
  # These radioButtons are used in the server as an input variable called
  # 'dataSource'
  radioButtons("dataSource", "",
               c("Data Source 2006" = "DataFile2006",
                 "Data Source 2011" = "DataFile2011")),
  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
        # This is a slider for the number of bins
        #The value determined here will be used as an input in server.R
        #The minimum is 1
        #The max is 50
        #It starts at a value of 30
        #by default it will grow by 1 but here I specify that it will step by 2
        #
        # The format is
        #sliderInput(VariableName, Stuff2WriteInTheWebPage, MinimumValue,MaxValue,Default,StepIncrement
        sliderInput("bins",  #This is the variable name, the value of which is determined by the slider
            "Number of bins:",#Label on the html page
            min = 1,#minimum for the slider
            max = 50,#maximum for the slider
            value = 30,#default value
            step=2),#slider increments

        # Question 1
        numericInput("minx", "x-axis minimum:", value = 1000),
        numericInput("maxx", "x-axie maximum:", value = 1000000),

        # Question 2
        checkboxInput("density", "Add density plot?", value = FALSE)),


    # Show a plot of the generated distribution
    mainPanel(
      # Question 3
      tabsetPanel(type = "tabs",
                  tabPanel("Population", plotOutput("distPlot1")),
                  tabPanel("Private Dwellings", plotOutput("distPlot2"))
                  )
#      plotOutput("distPlot")
    )

))
)
```

server.R

```
#
# This is the server logic of a Shiny web application. You can run the
# application by clicking 'Run App' above.
#
# Find out more about building applications with Shiny here:
#
#    http://shiny.rstudio.com/
#

library(shiny)

# Define server logic required to draw a histogram
shinyServer(function(input, output) {

  output$distPlot1 <- renderPlot({

    #Use those radioButtons to select the data to use
    #Note that the input was called "dataSource" in the ui.R file
    #Here it is an element from a list.
    #I extract it using input$nameOfThing
    if (input$dataSource == "DataFile2006") {
      x = read.csv("pop2006.csv")
      main = "Histogram of Canada population 2006" # Question 1
    } else {
      if(input$dataSource == "DataFile2011"){
        x = read.csv("pop2011.csv")
        main = "Histogram of Canada population 2011" # Question 1
      }else{
        #This means do not use any data.  This will break things.
        x=NULL
      }
    }
    #I didn't need to use if{}else{if{}else{}}
    # but I do so to show how to use multiple if else statements.


    # generate bins based on input$bins from ui.R
    x    <- x[-1, 3]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)
```

```r
#    # draw the histogram with the specified number of bins
#    hist(x, breaks = bins, col = 'darkgray', border = 'white',
#         xlab = "Population", # Question 1
#         main = main)         # Question 1


    # Question 2 density
    if (input$density) {
      hist(x, breaks = bins, col = 'darkgray', border = 'white',
           xlab = "Population", # Question 1
           main = main,         # Question 1
           probability = TRUE)  # Question 2

      dens <- density(x)
      lines(dens, col = "red")
    }else
    {
      hist(x, breaks = bins, col = 'darkgray', border = 'white',
           xlab = "Population", # Question 1
           main = main)         # Question 1
    }




  })

  # Question 3
  output$distPlot2 <- renderPlot({
    if (input$dataSource == "DataFile2006") {
      x = read.csv("pop2006.csv")
      main = "Histogram of Private Dwellings in 2006"
    } else {
      if(input$dataSource == "DataFile2011"){
        x = read.csv("pop2011.csv")
        main = "Histogram of Private Dwellings in 2011"
      }else{
        x=NULL
      }
    }

    x    <- x[-1, 4]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    if (input$density) {
      hist(x, breaks = bins, col = 'darkgray', border = 'white',
           xlab = "private dwelling",
           main = main,
           probability = TRUE)

      dens <- density(x)
      lines(dens, col = "red")
    }else
    {
      hist(x, breaks = bins, col = 'darkgray', border = 'white',
           xlab = "private dwelling",
           main = main)
    }

  })


})
```

output screenshots



D:/Simon Fraser University/2021 spring/STAT 240/Labs/lab7/shiny - Shiny

http://127.0.0.1:3161 | Open in Browser

Population Data

● Data Source 2006
○ Data Source 2011

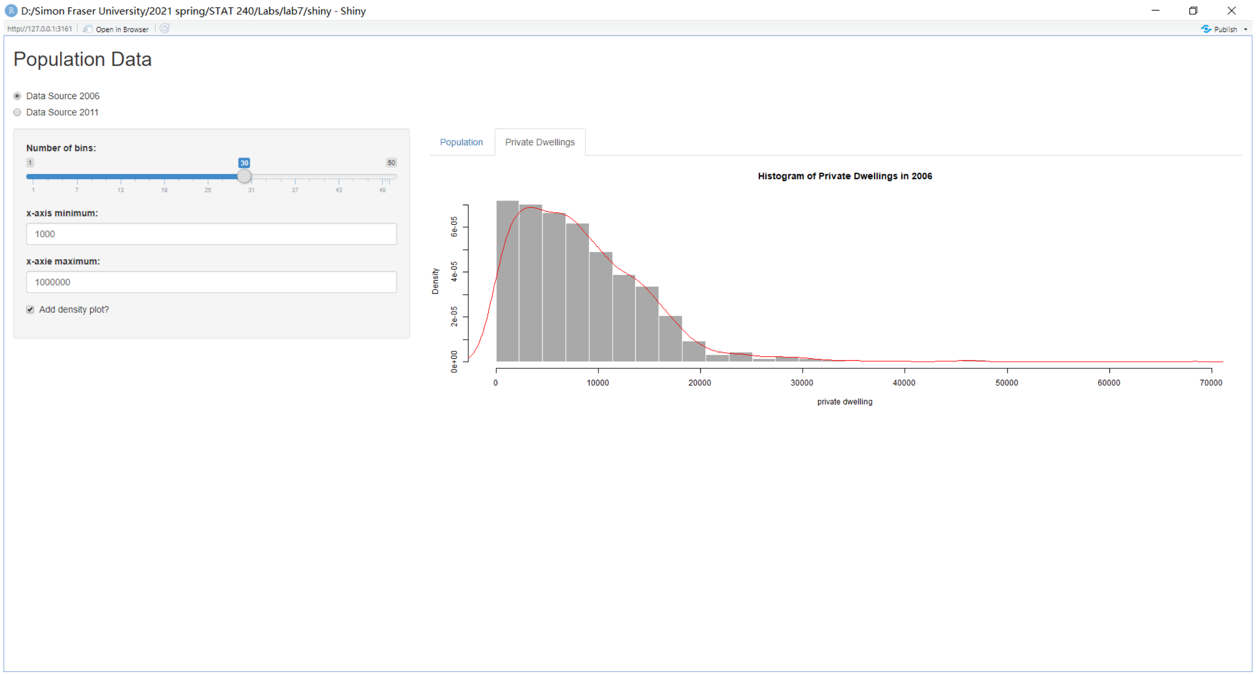
Number of bins: 30

x-axis minimum: 1000

x-axie maximum: 1000000

□ Add density plot?

Figure 1: Private Dwellings



D:/Simon Fraser University/2021 spring/STAT 240/Labs/lab7/shiny - Shiny

http://127.0.0.1:3161 | Open in Browser

Population Data

● Data Source 2006
○ Data Source 2011

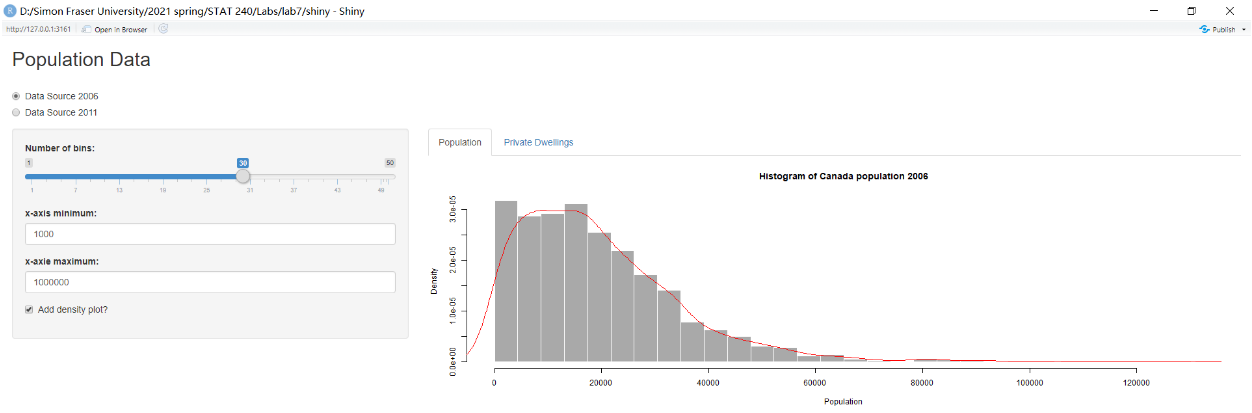
Number of bins: 30

x-axis minimum: 1000

x-axie maximum: 1000000

☑ Add density plot?

Figure 2: Private Dwellings with density estimate



D:/Simon Fraser University/2021 spring/STAT 240/Labs/lab7/shiny - Shiny

http://127.0.0.1:3161 | Open in Browser

Population Data

● Data Source 2006
○ Data Source 2011

Number of bins: 30

x-axis minimum: 1000

x-axie maximum: 1000000

☑ Add density plot?

Figure 3: Population tab