Intro to Data Science - HW 10 - Shiny


```
# Enter your name here: Victoria Haley
```

Attribution statement: (choose only one and delete the rest)

```
# 2. I did this homework with help from the book and the professor and these Internet so
urces: https://stackoverflow.com/questions/55070411/shinyapp-scatterplot-displays-only-o
ne-point
```

At a high level, we want to create a scatter plot, where the user can select what is the x axis and the y axis of the scatter plot. We will use the same dataset as used in our first ggplot homework.

The size and color will be represented by the year attribute, and the user will be able to choose one of the following (for the x and the y axis): new_sp_m014 new_sp_f014 new_sp_m65 new_sp_f65

Read in the same dataset we used for the ggplot homework

The file is: “https://intro-datascience.s3.us-east-2.amazonaws.com/who.csv (https://intro-datascience.s3.us-east-2.amazonaws.com/who.csv)” and store it in the *tb* dataframe

```
library(ggplot2)
library(tidyverse)
tb <- read.csv("https://intro-datascience.s3.us-east-2.amazonaws.com/who.csv")
```

Clean up the dataset, just like we did in the ggplot homework.

First, remove na’s from iso2

```
tb <- tb %>%
  filter(!is.na(iso2))
```

Now create the dataframe ‘tbCan’, which is the iso2 for canada (CA)

```
tbCan <-tb %>%
  filter(iso2 == "CA")
```

We will need the imputeTS package (only install if needed)

```
#no need for me to install.packages('imputeTS'), I already have it
library(imputeTS)
```

Now we can use 'na_interpolation for new_sp_m014

```
na_interpolation(tbCan$new_sp_m014)
```

```
##  [1] 12  8  6  9  3 11  9  9  4 10  3  7  6  8  2  1  3  0  4  0  5  6  0  1  2
## [26]  3  2  5  2
```

Create the User Interface of our shiny app

Define the sidebarPanel, which is two choices (use 'selectInput'), one for the x axis of the scatter plot, and the other is the y axis for the scatter plot. (make sure to library shiny).

```
library(shiny)
sidePanel <- sidebarPanel(
  #your code goes here
  selectInput(inputId = "x",
              label = "which x attribute to explore",
              choices = c("Males 0-14" = "new_sp_m014",
                          "Females 0-14" = "new_sp_f014",
                          "Males 65+" = "new_sp_m65",
                          "Females 65+" = "new_sp_f65")),
 selectInput(inputId = "y",
              label = "which y attribute to explore",
              choices = c("Males 0-14" = "new_sp_m014",
                          "Females 0-14" = "new_sp_f014",
                          "Males 65+" = "new_sp_m65",
                          "Females 65+" = "new_sp_f65")))
```

Create the mainPanel to show the scatter plot.

```
mainPanel <-  mainPanel(
    # your code goes here
      plotOutput("distPlot")
      )
```

Define UI for application (called ui)

Use a 'fluidPage' using the sidebarLayout, and your sidePanel and mainPanel

```
ui <- fluidPage(

    # Application title
    titlePanel("Exploring Data"),

    # Sidebar with a slider input for number of bins
    sidebarLayout(
        sidebarPanel(
          selectInput(inputId = "x",
              label = "which x attribute to explore",
              choices = c("Males 0-14" = "new_sp_m014",
                          "Females 0-14" = "new_sp_f014",
                          "Males 65+" = "new_sp_m65",
                          "Females 65+" = "new_sp_f65")),
        selectInput(inputId = "y",
              label = "which y attribute to explore",
              choices = c("Males 0-14" = "new_sp_m014",
                          "Females 0-14" = "new_sp_f014",
                          "Males 65+" = "new_sp_m65",
                          "Females 65+" = "new_sp_f65")),
        ),

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

Now let's define the server

Use ggplot to render a scatter plot, using the tbCan dataframe, the input for the x-axis and the input for the y-axis.

Store the results in the 'server' variable

```
server <- function(input, output) {

    output$distPlot <- renderPlot({
        #code to generate the scatter plot goes here
      ggplot(tbCan, aes_string(x=input$x, y=input$y)) +
geom_point(aes(size=year, color=year))
    })
}
```

Now run the shiny App

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

Exploring Data

which x attribute to explore

Males 0-14

which y attribute to explore

Females 0-14