Interactive plotting? plotly OR shiny

Yuexiao Pan, yp2489; Xishi Chen, xc2455

The packages we are going to use in the post shown as the following:

```
library(GDAdata)
library(tidyverse)
library(plotly)
library(shiny)
```

Introduction

When doing the explortary data visualization in R, people are not just satisfied with static plots. It is usually to start with static plots using basic R plots or `ggplot2` , however, interactive plots come to the stage at users' request of the ability of handling multiple wide data columns with various choices. Both `plotly` and `shiny` packages allow you to create beautiful, reactive D3 plots that are particularly powerful in websites and dashboards. In the post, we are going to compare those two packages using the R built-in data set *Speedski* in `GDAdata` package.

A general view for *Speedski* data set

```
head(SpeedSki)
```

```
##   Rank Bib FIS.Code                 Name Year Nation  Speed  Sex     Event
## 1    1  61     7039       ORIGONE Simone 1979    ITA 211.67 Male Speed One
## 2    2  59     7078         ORIGONE Ivan 1987    ITA 209.70 Male Speed One
## 3    3  66   190130       MONTES Bastien 1985    FRA 209.69 Male Speed One
## 4    4  57     7178 SCHROTTSHAMMER Klaus 1979    AUT 209.67 Male Speed One
## 5    5  69   510089         MAY Philippe 1970    SUI 209.19 Male Speed One
## 6    6  75     7204          BILLY Louis 1993    FRA 208.33 Male Speed One
##   no.of.runs
## 1          4
## 2          4
## 3          4
## 4          4
## 5          4
## 6          4
```

```
str(SpeedSki)
```

```
## 'data.frame':    91 obs. of  10 variables:
##  $ Rank      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Bib       : int  61 59 66 57 69 75 67 58 62 56 ...
##  $ FIS.Code  : int  7039 7078 190130 7178 510089 7204 7053 7170 7230 7055 ...
##  $ Name      : Factor w/ 91 levels "ABRAHAMSSON Mats",..: 64 63 56 83 54 13 66 14 19 43 ...
##  $ Year      : int  1979 1987 1985 1979 1970 1993 1975 1991 1980 1982 ...
##  $ Nation    : Factor w/ 14 levels "AUT","BEL","CAN",..: 7 7 5 1 12 5 13 5 4 13 ...
##  $ Speed     : num  212 210 210 210 209 ...
##  $ Sex       : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Event     : Factor w/ 3 levels "Speed Downhill",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ no.of.runs: int  4 4 4 4 4 4 4 4 4 4 ...
```

We mainly use the following three variables to make a histogram plot and a scatterplot in comparsion of `plotly` and `shiny` :

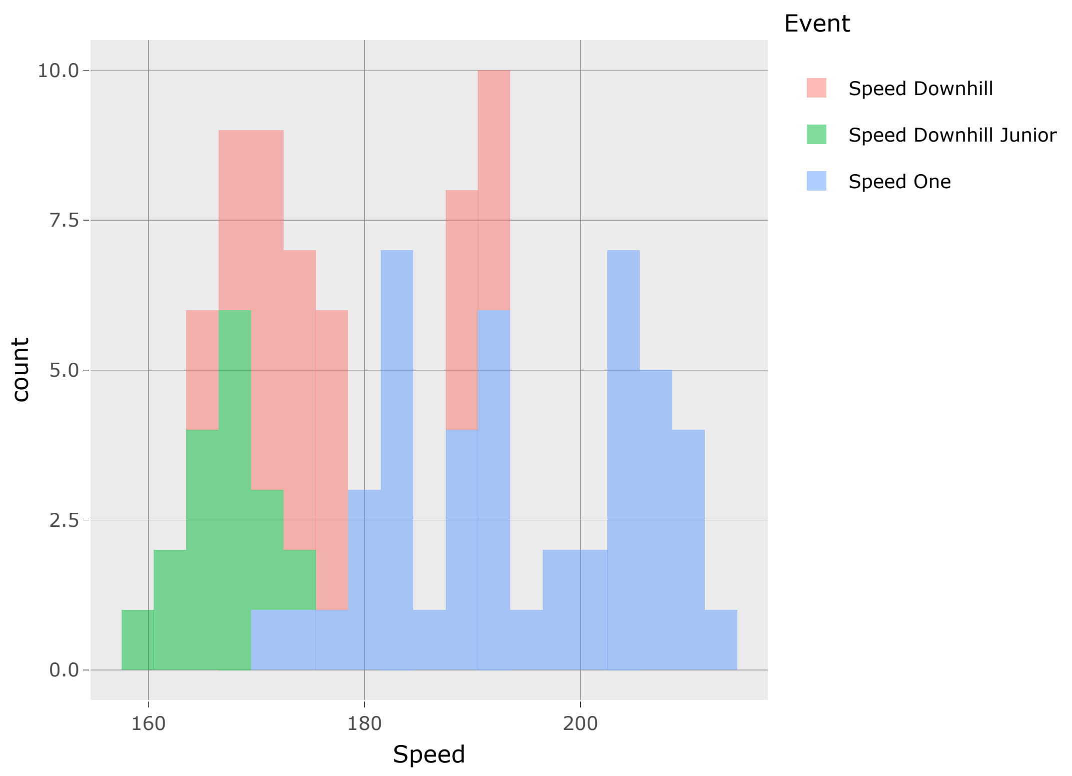
- **Year**(integer): Skier's year of birth, ranging from 1952 to 1995
- **Speed**(numeric): Speed achieved in km/hr, ranging from 160 to 212
- **Sex**(factor): Female / Male

Example using `plotly`

We plot a histogram for variable `Speed` and fill the plot according to variable `Event` . From the plot we can see that athletes' speed in *Speed one* are usually higher than that in *Speed Downhill Junior*.

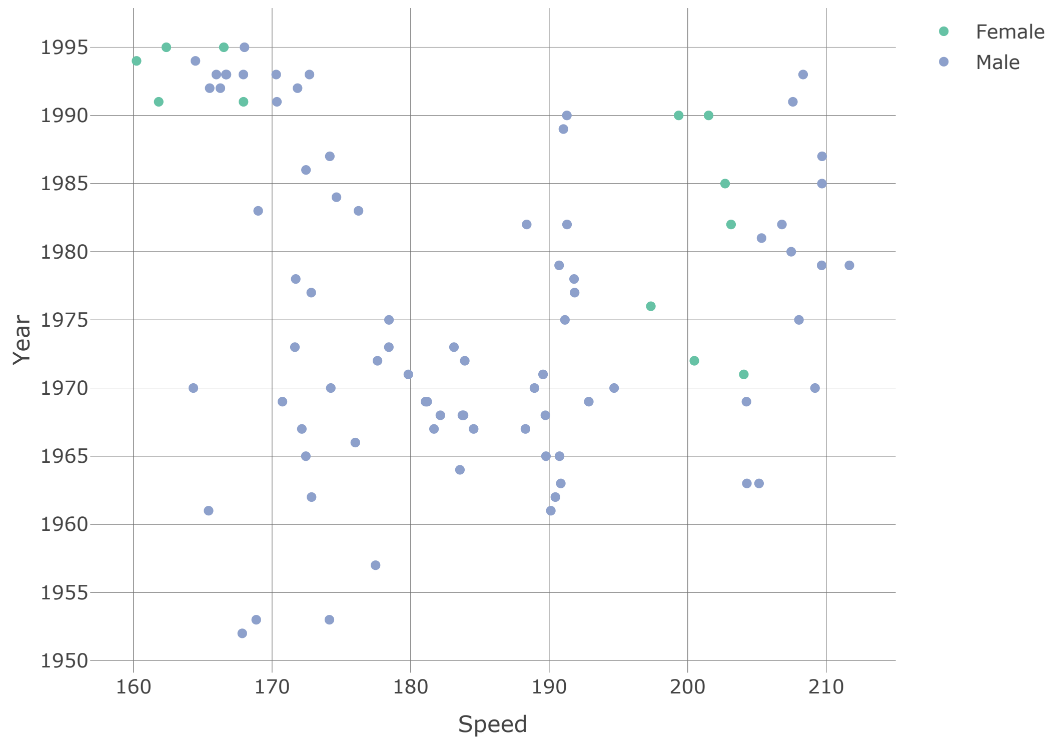
When you move your mouse over the plot, there is a floating window shown with the same filled color as the area you click on. It gives you the all the data information about the point (i.e. in the example, it shows *Speed*, *Event* and *count* of that point).

```
g <- ggplot(SpeedSki)+
  geom_histogram(aes(x = Speed, fill = Event),
                 alpha = 0.5,
                 binwidth = 3)

ggplotly(g)
```



In the following picture, we generate a scatter plot of *Speed* and *Year* and categorize data with gender. In this dataset, there are more males than females. The points for male spread out the scatter plot while the points for females cluster into two groups. Now the floating window shows the corresponding coordinate values (i.e. (Speed, Year)) and gender when your mouse moving over each point in the plot.

```
plot_ly(SpeedSki, x = ~Speed, y = ~Year,
        type = 'scatter', mode = 'markers', color = ~Sex)
```



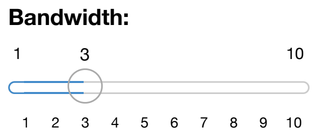
There are other options shown in the top right of each plot when your mouse moving around. For example, you can zoom in/out the plot, drag certain area you want to explore more and even download the plot. You are welcome to explore about the `plotly` but here we focus on the package comparison.

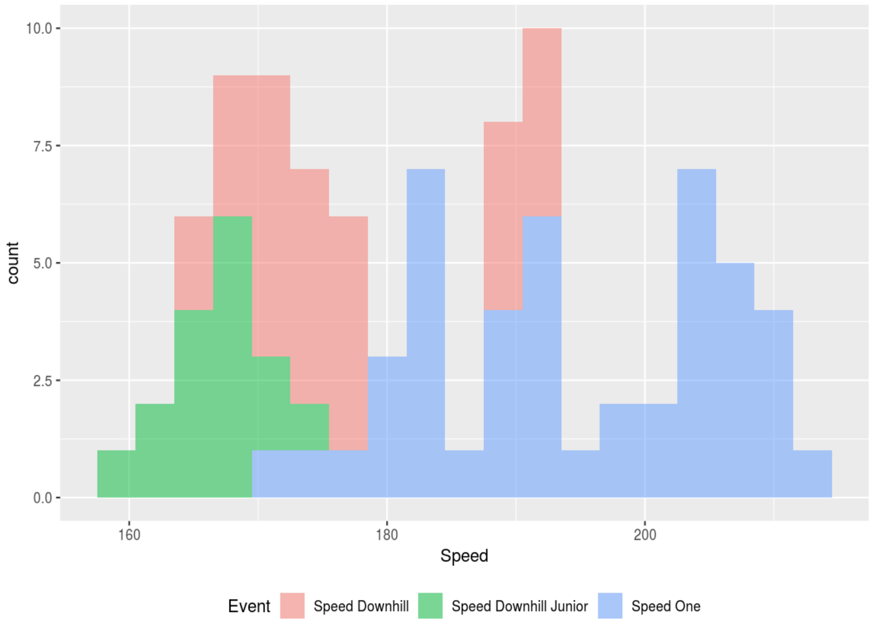
Example using `shiny`

In shiny app, we design some user interactions. For the histogram, users could choose different bandwidths. In the scatter plot, two inputs could be changed by users: Year rang and gender. There is also a horizontal line which shows the upper bound of the year rang in the plot.

[scroll down to explore more]

```
ui <- fluidPage(
  fluidRow(
    column(3,
      sliderInput("bw", label = "Bandwidth:",
                min = 1, max = 10,value=3, step = 0.2)
    ),
    column(8,
      plotOutput('plot')
    )
  ),

  fluidRow(
    column(3,
      checkboxGroupInput("gender", label = "Gender:",
                  choices = c("Female", "Male"), selected = c("Male","Female")),

      sliderInput("year", label = "Year Rang:",
                  min = 1950, max = 2000, value = 2000, step = 1)
    ),
    column(8,
      plotOutput('pplot')
    )
  )
)

server <- shinyServer(function(input, output) {
  output$plot=renderPlot({
    ggplot(SpeedSki)+
      geom_histogram(aes(x=Speed,fill=Event),alpha=0.5,binwidth=as.numeric(input$bw))+
      theme(legend.position="bottom")
  })
  output$pplot=renderPlot({
    SpeedSki[SpeedSki$Sex == input$gender & SpeedSki$Year <= input$year,] %>%
      ggplot(aes(Speed, Year, color = Sex)) + geom_point() +
        xlim(c(158, 212)) + ylim(c(1950, 2000)) +
        geom_hline(yintercept=input$year) +
        theme(legend.position="bottom")
  })
})

shinyApp(ui,server)
```





Comparison of plotly and Rshiny

Generally speaking, both `plotly` and `shiny` can be used to creat interactive plots.

Both of them have their own communities (plotly community (https://plot.ly/r/getting-started/), Shiny community (https://shiny.rstudio.com/articles/)) and are actively under development. `plotly` is a transformable package, which can be used in R, python, Matlab and other languages. `shiny` is built in R studio, mainly created to design a hands on web application without knowing prounding knowlegde such HTML or Javascript.

Docummentation in R help is more detailed in `plotly` and `plotly` syntax is very intuitive.

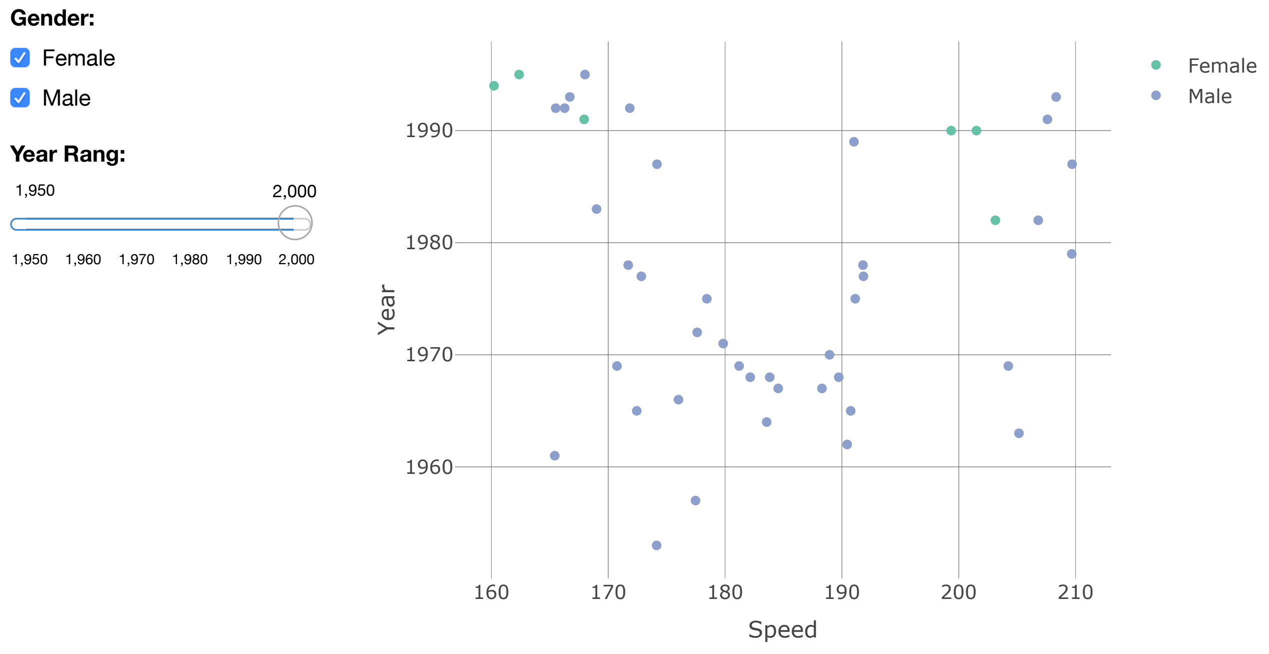
The most noticeable difference in the examples shown above is that `plotly` allows users to hover the mouse over the plots and see the exact data values, zoom in and out of specific regions, and capture stills, while `shiny` allows users to customise the plots, providing the options you made for them (like binwidth adjustment in histogram and variable range adjustment in scatterplot).

Conclusion

`plotly` is quicker to build with a better syntax and deatiled documentation. It is easier to get a basic interactive plot, which provides direct insights on how you want to further work on it. `shiny` is more complicated and requires more efforts and layers to build an interactive dashboard. Actually, `plotly` can be used in `shiny` , through which you can realize to design an interactive dashboard not only with customised options but also detailed infomation. Both of them are powerful, serving different purposes. It is recommonded to use `plotly` in RMarkdown and further visualize with `shiny` , or design web apps in both `shiny` and `plotly` .

The following simple example shows that how to embed plotly graphs in shiny, with the *SpeedSki* scatter plot:

```r
ui2 <- fluidPage(
  fluidRow(
    column(3,
      checkboxGroupInput("gender", label = "Gender:",
                  choices = c("Female", "Male"), selected = c("Male","Female")),

      sliderInput("year", label = "Year Rang:",
                  min = 1950, max = 2000, value = 2000, step = 1)
    ),
    column(8,
      plotlyOutput('pplot')
    )
  ),
  verbatimTextOutput("event")
)

server2 <- shinyServer(function(input, output) {
  output$pplot=renderPlotly({
    SpeedSki[SpeedSki$Sex == input$gender & SpeedSki$Year <= input$year,] %>%
      plot_ly(x = ~Speed, y = ~Year, type = 'scatter', mode = 'markers', color = ~Sex)
  })

  output$event <- renderPrint({
    event_data("plotly_hover")
  })

})

shinyApp(ui2,server2)
```


Gender:
Female
Male
Year Rang:
1,950
2,000
1,950 1,960 1,970 1,980 1,990 2,000
Year
1990
1980
1970
1960
160
170
180
190
200
210
Speed
Female
Male