Short Tutorial of Shiny

Shiny is a recently developed tool to build interactive web applications based on R (first released in Nov. 2012).

Let's see some examples first:

http://glimmer.rstudio.com/wisconsindpi/distribution/

http://glimmer.rstudio.com/wisconsindpi/correlation/

http://glimmer.rstudio.com/winston/heightweight/

http://glimmer.rstudio.com/systematicin/retirement.withdrawal/

http://glimmer.rstudio.com/qbrc/grn/

http://glimmer.rstudio.com/timelyportfolio/shiny-d3-showreel/

Why is Shiny?

1. Zero knowledge of HTML/JS/CSS is required, but fully extensible with them!
2. Extend Interactivity to Reactivity: **Reactive Programming!**

To build a simple Shiny application, one needs:

- **a user interface file(UI)**—define the client part with inputs and outputs;
- **a server file**—define the task for the server part given inputs and outputs;

  and other data, .global file, .html file, etc, that support the application.

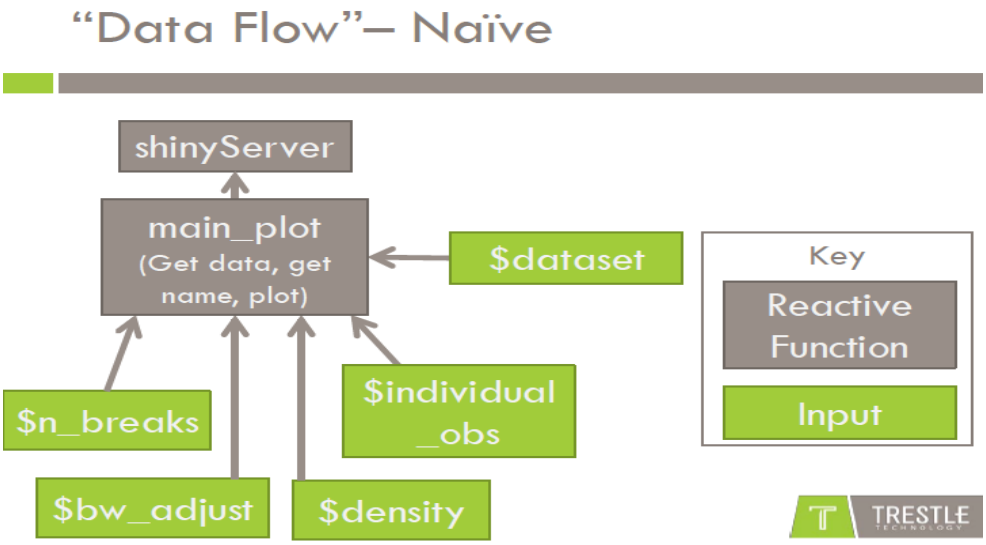

Figure is quoted from Jeff Allen's tutorial on

http://www.r-bloggers.com/introduction-to-shiny-slides/

Task One: Compare several match facts between winners and losers with a boxplot.

- **Step one:** create a folder including the "USOP.csv" data. (Call the folder "shiny")
- **Step two:** install Shiny and googleViz, create a ui.R file and another server.R file in this folder.

```
install.packages("shiny")
install.packages("googleViz")#This will be used in Task Two later.
```

- **Step three:** specify the ui.R file.

```
library(shiny)
shinyUI (pageWithSidebar
        (       #Specify Application title
                headerPanel ("Differences Between Winners and Losers"),
                #Sidebar with controls to select the variable to plot against match result
                sidebarPanel
                (       selectInput ("variable", "Variable:",
                                        list("Winner" = "winner",
                                "Error" = "error",
                                "Ace" = "ace",
                                        "Double Fault"="double")
                                ),
                        # Add an optional input: to specify whether outliers should be displayed
                        checkboxInput ("outliers", "Show outliers", FALSE)
                ),
                #Show the caption and plot of the requested variable against match result as outputs
                mainPanel
                (       h3(textOutput("caption")),
                        plotOutput("tennisPlot")
                )
        )
)
```

Step four: Specify the server file

```
library(shiny)
library(datasets)
library(foreign)
tennis <-read.csv("USOP.csv") # Why do we put data here instead of in the ui.R file?
# Define server logic required to plot various variables against match result
shinyServer(function(input, output)
{       # Construct the formula for the title of the plot
        formulaText <- reactive(
        {       paste(input$variable, "against match results")
        }
                                )
        #Return the formula text for printing as a caption
        output$caption <- renderText (
        {       formulaText()
        }
                                        )
          # Generate a boxplot of requested variable against result and include outliers if requested
        output$tennisPlot <- renderPlot(
        {       #Construct a formula for the plot
                boxplot(as.formula(paste(input$variable, "~Result" )),
                data = tennis,
                outline = input$outliers
                col="green")
        }
                                        )
}
            )
```

- **Step five:** Run the app. in R!

```
#Before running the app, we need to set the working directory to the location of the folder "shiny".
setwd("location of shiny app. folder")
runApp()
```

Task Two: Create a Geomap of the Nationalities of Players

- **Create the ui.R file:**

```
require(shiny)
        shinyUI(pageWithSidebar(
                headerPanel("Geo Chart of US Open Matches"),
                sidebarPanel(
                                sliderInput("Year", "Years of Matches to be Displayed:",
                                min=2003, max=2011, value=2011,step=1,
                                format="###0",animate=TRUE)
                                 ),
                mainPanel(  h3(textOutput("year")),
                                 htmlOutput("gvis")
                            )
                                    )

              )
```

- **Create the server.R file:**

```
require(googleVis)
require(shiny)
#Prepare data to be displayed, load "USOP.csv"
tennis <- read.csv("USOP.csv")
attach(tennis)
#Do a data structure transformation in order to make the data recognized by googleViz
#Use table() to generate a count table and form a new data frame with “country”,”year”&”count”
tab<-table(year,country)
con<-rep(colnames(tab),each=length(rownames(tab)))
yea<-rep(rownames(tab),length(colnames(tab)))
counts<-tab[1:length(tab)]
#Merge these new created vectors into a new data frame
data<-data.frame(country=con,year=yea,count=counts)
```

```
#Reactive part of sever.R

shinyServer(function(input, output) {

        #How many inputs and outputs do we have here?

        myYear <- reactive({ input$Year })

        output$year <- renderText({

                paste("The Matches of US Open Played in the Year of", myYear())

                        })

        output$gvis <- renderGvis({

                myData <- subset(data, year==myYear())

                #Visualization with gvisGeochart() from the googleViz package.

                gvisGeoChart(myData, locationvar="country", colorvar="count",

                options=list(region="world", width=1000, height=700,

                    colorAxis="{colors:['white','pink','magenta', 'purple']}"

                        ))

                        })

                })
```

Useful materials for Shiny:

http://www.mytinyshinys.com/index.html

http://cran.r-project.org/web/packages/shiny/shiny.pdf

http://cran.r-project.org/web/packages/googleVis/index.html (for googleViz)

http://rstudio.github.io/shiny/tutorial/

Search for Shiny in Github, folk the following useful repositories:

https://github.com/trestletech/shiny-sandbox

https://github.com/rstudio/shiny