shinyscript

Jessica Cole

November 28, 2018

# Shiny App  
# November 28 2018  
# JSC  
  
# shinyApp is r package that builds interactive web apps( not page).  
# uses R coding and interactiveness of web to create a reactive experience.  
# ui : user interface: this is the controls and layout/appearance of the app. This is what the user will see.   
# server: the coding instructions for the shiny app  
# control widgets: web elements that users will interact with.   
  
# library(shiny)  
# #Define UI ----  
# ui <- fluidPage(  
# titlePanel(""),  
# sidebarLayout(  
# sidebarPanel(  
# ),  
# mainPanel(  
# )  
# )  
# )  
#   
# #Define server ----  
# server <- function(input,output){  
# }  
#   
# #Run the app ----  
# shinyApp(ui=ui,server = server)  
   
  
library(shiny)  
  
  
ui <- fluidPage(  
   
 # add a title to shiny App  
 titlePanel("This is a Test shiny"),  
   
 # create sidebar for interactive elements  
 sidebarLayout(position ="left",  
   
 # add sidebar panel  
 sidebarPanel(  
   
 # add a header in side panel  
 h1("This is my first Header"),  
 h2("second"),  
 h3("third"),  
 # add in widgets. widgets allow user to change elements  
 # selectInput requires: name, label, choices)  
 selectInput("X",label = "X", names(trees)),  
 selectInput("Y", label = "Y", names(trees))),  
 mainPanel(  
   
 # add a header in main panel as well  
 h1("header"),  
   
 # bold text  
 h2(strong("Header 2")),  
   
 # add a paragraph  
 p("This is a paragraph about my graph. It's a plot of how many x are in a square. Use this to give a description of what will be shown"),   
   
 # add line breaks  
 br(),  
   
   
 # add an image  
 img(src = "ant.jpg",height = 75, width = 75),   
   
 # insert the "output", the main reactive element  
 plotOutput("TreePlot")  
 )  
 )  
)  
  
#   
server <- function(input,output){  
  
 # combine the selected variables into new data frame  
 selectedData <- reactive({  
 trees[,c(input$X,input$Y)]  
 })  
   
 # use render plot to add to reactive element  
 output$TreePlot <- renderPlot({  
   
 # basic plot function thats built into r  
 plot(selectedData(),  
   
# type refers how the data is shown: p - points, l - lines, b - both  
 type = "p",  
  
# change point:using pch values stored in 'plot' function. 21.25 - color and fill points, 19 - solid circle, only need col for color  
 pch = 21.25,   
 col = "black",  
 bg = "blue",  
  
# add a main title to graph  
 main = "This is the main title of the Graph",  
  
# add a subtitle as a caption  
 sub = "this is the subtitle for the plot. A good way to add a caption below the graph")  
   
 }  
 )  
}  
   
# call to shiny app  
shinyApp(ui=ui,server=server)

Shiny applications not supported in static R Markdown documents