server.R

ybw5116

Wed May 29 14:10:36 2019

library(shiny)  
library(treemap)  
library(RColorBrewer)  
library(ggplot2)  
library(shinyBS)  
library(shinyWidgets)  
  
##Weight adjustment   
  
shinyServer(function(input, output,session) {  
   
 observeEvent(input$info,{  
 sendSweetAlert(  
 session = session,  
 title = "Instructions:",  
 text = "Move the sliders to explore how the weighting adjustment affects the results.",  
 type = "info"  
 )  
 })  
 observeEvent(input$info1,{  
 sendSweetAlert(  
 session = session,  
 title = "Instructions:",  
 text = "Move the sliders to explore how the weighting adjustment affects the results.",  
 type = "info"  
 )  
 })  
 ###UPDATE: adding the go button   
 observeEvent(input$go,{  
 updateTabItems(session,"tabs","easy")  
 })  
 observeEvent(input$start,{  
 updateTabItems(session,"tabs","overview")  
 })  
   
 ##############################read in the dataset################################################  
 originaldata <- read.csv("originalDataset.csv", TRUE, sep = ",",na.strings = TRUE)  
 dataf = data.frame(originaldata)  
 population <- read.csv("population.csv",TRUE, sep = ",", na.strings = TRUE)  
 datafP = data.frame(population)  
 sample <- read.csv("sample.csv", TRUE, sep = ",",na.strings = TRUE)  
 datafS = data.frame(sample)  
 electionPopulationEW <- read.csv("electionPopulationRace.csv",TRUE, sep = ",", na.strings = TRUE)  
 eleDatafEW = data.frame(electionPopulationEW)  
   
   
 output$dataTable <- renderTable(read.csv("dataTable.csv"))  
 output$populationRatio <- renderTable(read.csv("PopulationRatio.csv"))  
   
   
   
 inputs= reactive({  
 value = c(input$male,input$female,input$white\*0.5,input$black\*0.2,input$hispanic\*0.15,input$asian\*0.10,input$other\*0.05)  
 })  
   
 #############################Easy level###########################################################  
   
   
 ###UPDATE###  
 #create the static bar chart for Easy level (sample and pop)  
 output$barPopSample <- renderPlot({  
 barTable <- matrix(c(52,70,48,30),ncol=2, byrow = TRUE)  
 colnames(barTable) <- c("Poplulation", "Sample")  
 barTable <- as.table(barTable)  
   
 barplot(barTable, main="Gender Proportion",  
 col=c("#FBB4AE","#B3CDE3"),  
 width = 0.8, xlim = c(0,2),cex.names=1, cex.main = 1.3)  
 par(lwd = 2)  
 legend("topright", c("Female","Male"), fill=c("#FBB4AE","#B3CDE3"))  
   
}, width = 370, height = 320)  
  
  
  
 ##create the interactive treemap for easy level  
 output$samplePop <- renderPlot({  
   
 value = inputs()  
 dataf[c(1,3),"Population"] = dataf[c(1,3),"Population"] \* value[1]  
 dataf[c(2,4),"Population"] = dataf[c(2,4),"Population"] \* value[2]  
   
 treemap(dataf, index=c("Gender","TVshow"), vSize = "Population", type="index",   
 palette = colorRampPalette(brewer.pal(4, "Pastel1"))(4),   
 title="Use Sample to Represent Population", fontsize.title = 14, fontsize.labels = 16)  
 }, width = 300, height = 260)  
   
   
 ##create the bar chart for Easy level  
 output$bar <- renderPlot({  
   
 value = inputs()  
 dataf[c(1,3),"Population"] = dataf[c(1,3),"Population"] \* value[1]  
 dataf[c(2,4),"Population"] = dataf[c(2,4),"Population"] \* value[2]  
   
 ellen = c(dataf[2,"Population"], dataf[1,"Population"])  
 late = c(dataf[4,"Population"], dataf[3,"Population"])  
 dataframe = data.frame("The Ellen Show" = ellen, "The Late Night Show" = late)  
 matrix = as.matrix(dataframe)  
 par(lwd = 2)  
 barplot(matrix, col = c("#FBB4AE","#B3CDE3"), main = "Supporting Rate of Both Show",   
 width = 0.8, xlim = c(0,2),cex.names=1, cex.main = 1.3, ylim = c(0,60))  
 par(lwd = 2)  
 legend("topright", c("Female","Male"), fill=c("#FBB4AE","#B3CDE3"))  
   
 }, width = 370, height = 320)  
   
 ################################Hard level##############################################################  
 ##horizontal bar plot  
 output$elePopEW <- renderPlot({  
 value = inputs()  
   
 barplot(prop.table(rbind(c(eleDatafEW[1,4],eleDatafEW[4,4],eleDatafEW[7,4],eleDatafEW[10,4],eleDatafEW[13,4]),  
 c(eleDatafEW[2,4],eleDatafEW[5,4],eleDatafEW[8,4],eleDatafEW[11,4],eleDatafEW[14,4]),  
 c(eleDatafEW[3,4],eleDatafEW[6,4],eleDatafEW[9,4],eleDatafEW[12,4],eleDatafEW[15,4])))  
 ,horiz = TRUE, col = c("#002868","azure1","#BF0A30"), names.arg = c("White","Black","Latino","Asian","Other")  
 , main = "Comparison of Two Candidates", las = 1  
 , width = c(value[3],value[4],value[5],value[6],value[7])  
 )  
 },width = 500, height = 300)  
   
 ##vertical bar plot  
 output$elePopWBar <- renderPlot({  
 value = inputs()  
  
 barplot(prop.table(rbind(c(eleDatafEW[1,4] \* value[3],eleDatafEW[3,4] \* value[3]),  
 c(eleDatafEW[4,4] \* value[4],eleDatafEW[6,4] \* value[4]),  
 c(eleDatafEW[7,4] \* value[5],eleDatafEW[9,4] \* value[5]),  
 c(eleDatafEW[10,4]\* value[6],eleDatafEW[12,4]\* value[6]),  
 c(eleDatafEW[13,4]\* value[7],eleDatafEW[15,4]\* value[7])))  
 , names.arg = c("Clinton","Trump")  
 , col= brewer.pal(8, "YlOrBr")  
 )  
 },width = 250, height = 300)  
   
#####################################################################################################   
 ##Hints  
 output$hintM <- renderText(  
 if (input$male == 1.4 | input$male == 1.8){print("You are close to the right answer.")}  
 else if (input$male == 1.6){print("Congratulations! You got the correct weight for male.")}  
 else {print("Move the slider to reach the correct weight.")}  
 )  
 output$hintF <- renderText(  
 if (input$female == 0.74){print("Congratulations! You got the correct weight for female.")}  
 else if (input$female >= 0.7 & input$female <= 0.8){print("You are close to the right answer.")}  
 else {print("Move the slider to reach the correct weight.")}  
 )  
 output$Congrats <- renderText(  
 print("Congrats!")   
 )  
  
 output$Solutions <- renderText(  
 print("Finding the correct weight is hard, especially when the population proportion is unknown. ")  
 )  
  
 ##progress bar  
 output$progress <- renderUI({  
 if (sum(round(input$male \* 30),round(input$female \* 70)) <= 100){  
 tags$div(  
 'class' = "progress progress-striped active",  
 tags$div('class' = "progress-bar progress-bar-info", 'style'=paste0("width:",round(input$male \* 30),"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-default", 'style'=paste0("width:",round(input$female \* 70),"%",sep = ''))  
 )  
 }else{  
 tags$div(  
 'class' = "progress progress-striped active",  
 tags$div('class' = "progress-bar progress-bar-info", 'style'=paste0("width:",round(input$male \* 30),"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-default", 'style'=paste0("width:",100 - round(input$male \* 30),"%",sep = ''))  
 )  
 }  
   
 })  
 output$progressB <- renderUI({  
 value = inputs()  
   
 if (sum(value[3], value[4], value[5], value[6], value[7]) <= 1){  
 tags$div(  
 'class' = "progress progress-striped active",  
 tags$div('class' = "progress-bar progress-bar-success", 'style'=paste0("width:",value[7] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-warning", 'style'=paste0("width:",value[6] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-danger", 'style'=paste0("width:",value[5] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-info", 'style'=paste0("width:",value[4] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-default", 'style'=paste0("width:",value[3] \* 100,"%",sep = ''))  
 )  
 }else{  
 tags$div(  
 'class' = "progress progress-striped active",  
 tags$div('class' = "progress-bar progress-bar-success", 'style'=paste0("width:",value[7] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-warning", 'style'=paste0("width:",value[6] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-danger", 'style'=paste0("width:",value[5] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-info", 'style'=paste0("width:",value[4] \* 100,"%",sep = '')),  
 tags$div('class' = "progress-bar progress-bar-default", 'style'=paste0("width:",(100 - sum(value[4], value[5], value[6], value[7])),"%",sep = ''))  
 )  
 }  
   
 })  
   
 ##Feedbacks  
 output$warning <- renderUI({  
 if (input$male \* 30 + input$female \* 70 <= 100){  
 h4("Notice that the summation bar should never be larger or smaller than one because the weighted sample should have the same sample size as the real sample.")  
 }else{  
 h4("Warning: The summation is now larger than n.",style = "color: red")  
 }  
 })  
 output$warningB <- renderUI({  
 value = inputs()  
   
 if (sum(value[3], value[4], value[5], value[6], value[7]) <= 1){  
 h4("Notice that the summation bar should never be larger or smaller than one because the weighted sample should have the same sample size as the real sample.")  
 }else{  
 h4("Warning: The summation is now larger than n.",style = "color: red")  
 }  
 })  
 output$Congradulation <- renderText(  
 print("Congratulations! This is the result of 2016 Exit Polls.")  
 )  
   
})