Nauka dla społeczeństwa

Katedra i Zakład Informatyki i Statystyki

dr Justyna Marcinkowska

data przygotowania: 2024-05-05

Zalecenia dotyczące warsztatów cz.2. – kody RShiny – cd.:

*Praktyka dla statystyka*

*BADANIA Z OBSZARU ZDROWIA*

*RShiny*

[R Shiny](https://raw.githubusercontent.com/rstudio/cheatsheets/main/shiny.pdf) to pakiet języka R do tworzenia interaktywnych aplikacji webowych. Aplikacje Shiny mogą być wykorzystywane do tego, by własne procedury statystyczne udostępniać  w przyjaznej formie badaczom nie posiadającym umiejętności programistycznych w języku R.

**Podczas warsztatów uczestnicy nauczą się:**

1. Tworzyć proste aplikacje Shiny
2. Dodawać elementy interaktywne do aplikacji
3. Wizualizować dane w aplikacji

# Kody do zadań z RShiny

Na początku każdego kodu ładujemy pakiety:

library(shiny)

library(rstatix)

library(dplyr)

library(ggplot2)

library(readxl)

library(plotly)

library(ggsci)

# zad.1

#"Warsztaty z RShiny PRAKTYKA DLA STATYSTYKA"
#cz.II
#zadanie 1 - interaktywny boxplot z kolorami skrzypiec

# tu library

sodium <- read\_excel("../sodium.xlsx",sheet = "data")

# w ui: dodano wyświetlenie w selectInpucie nazw polskich
# w serverze: dodano warstwę skrzypiec z przypisanym kolorem wypełnienia
# wg. zmiennej grupującej (tu: department)

# Define UI for application that draws a histogram
ui <- fluidPage(

 # Application title
 titlePanel("tytuł...."),

 # Sidebar with a slider input for number of bins
 selectInput("zmienna","Wybierz zmienną numeryczną",
 choices = c("Wiek" = "age",
 "Stężenie sodu" = "sodium\_concentration",
 "Pobyt w oddziale ratunkowym" = "Stay\_in\_the\_Emergency\_Department"),
 selected = "age"),

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

)

# Define server logic required to draw a histogram
server <- function(input, output) {

 output$wykres <- renderPlot({
 sodium %>%
 filter(hospitalization==1) %>%
 ggplot(aes(x=department,y=!!sym(input$zmienna)))+
 geom\_violin(aes(fill=department))+
 geom\_boxplot()
 })
}

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

# zad.2

#"Warsztaty z RShiny PRAKTYKA DLA STATYSTYKA"
#cz.II
#zadanie 2 - interaktywny boxplot + typ wykresu do wyboru na radioButtons()

# tu library

# W tej aplikacji dodajemy radioButtons aby wybierać rodzaj wykresu
# 1.) modyfikujemy ui:
# po selectInpucie: radioButtons("czy\_box","Typ wykresu: ",c("box plot"="b","violin"="v"))
# 2.) modyfikujemy server:
# dodajemy if()else sprawdzający czy\_box i rysujący jeden lub drugi wykres

 sodium <- read\_excel("../sodium.xlsx", sheet = "data")
 sub\_sodium<- sodium %>%
 filter(hospitalization==1)

 pal <- pal\_jco() # przypisanie palety do zmiennej

 ui <- fluidPage(

 # Application title
 titlePanel("Pierwszy wykres interaktywny"),
 # Sidebar with a slider input for number of bins
 sidebarLayout(
 sidebarPanel(
 wellPanel(style = "background: darkgray",
 selectInput("zmienna", "Wybierz zmienną:",
 choices = c("Wiek" = "age",
 "Stężenie sodu" = "sodium\_concentration",
 "Pobyt w oddziale ratunkowym" = "Stay\_in\_the\_Emergency\_Department"),
 selected = "age")
 ),
 radioButtons("czy\_box","Typ wykresu: ",c("box plot"="b","violin"="v"))
 ),
 mainPanel(
 wellPanel(style = "background: slateblue",plotOutput("wykres"))
 )))

 server <- function(input, output) {
 output$wykres <- renderPlot({

 if(input$czy\_box=="b"){
 sub\_sodium %>%
 ggplot( aes(x=department,y=!!sym(input$zmienna))) +
 geom\_boxplot(aes(fill = department))+
 scale\_fill\_jco()+
 theme\_classic()
 }else
 {sub\_sodium %>%
 ggplot( aes(x=department,y=!!sym(input$zmienna))) +
 geom\_violin(aes(fill = department))+
 scale\_fill\_jco()+
 theme\_classic()
 }
 })
 }
 #Run the application
 shinyApp(ui = ui, server = server)

# zad.3

#"Warsztaty z RShiny PRAKTYKA DLA STATYSTYKA"
#cz.II
#zadanie 3:
# interaktywny boxplot z radioButtonem zamienionym na check + zmienna grupująca,
# czyli nowy selectInput()

# tu library

 sodium <- read\_excel("../sodium.xlsx", sheet = "data")
 sub\_sodium<- sodium %>%
 filter(hospitalization==1)

 sub\_sodium$medical\_transport <- factor(sub\_sodium$medical\_transport,
 levels = c(0, 1), labels = c("nie", "tak"))

 pal <- pal\_jco() # przypisanie palety do zmiennej

 ui <- fluidPage(

 # Application title
 titlePanel("Pierwszy wykres interaktywny"),
 # Sidebar with a slider input for number of bins
 sidebarLayout(
 sidebarPanel(
 wellPanel(style = "background: darkgray",
 selectInput("zmienna", "Wybierz zmienną:",
 choices = c("Wiek" = "age",
 "Stężenie sodu" = "sodium\_concentration",
 "Godziny pobytu na SOR" = "Stay\_in\_the\_Emergency\_Department"),
 selected = "age"),
 selectInput("zmienna\_gr", "Wybierz grupującą:",
 choices = c("Płeć" = "sex",
 "Oddział szpitalny" = "department",
 "Czy dowieziony transportem medycznym?" = "medical\_transport"),
 selected = "sex")
 ),
 checkboxInput("czy\_box","Czy wykres box? ",value=TRUE)
 ),
 mainPanel(
 wellPanel(style = "background: slateblue",plotOutput("wykres"))
 )))

 server <- function(input, output) {
 output$wykres <- renderPlot({

 if(input$czy\_box){
 sub\_sodium %>%
 ggplot( aes(x=!!sym(input$zmienna\_gr),y=!!sym(input$zmienna))) +
 geom\_boxplot(aes(fill = !!sym(input$zmienna\_gr)))+
 scale\_fill\_jco()+
 theme\_classic()
 }else
 {sub\_sodium %>%
 ggplot( aes(x=!!sym(input$zmienna\_gr),y=!!sym(input$zmienna))) +
 geom\_violin(aes(fill = !!sym(input$zmienna\_gr)))+
 scale\_fill\_jco()+
 theme\_classic()
 }
 })
 }
 #Run the application
 shinyApp(ui = ui, server = server)