Case Study 4

AKSTA Statistical Computing

Fani Sentinella-Jerbić

07.04.2023

Data Preparation

I'm changing the column names directly in the dataframe to make it easier later.

```r
library(ggplot2)
library(dplyr)
library(countrycode)
load(file = "data_cia.rda")
world_map <- map_data("world")
world_map$iso2c <- countrycode::countrycode(sourcevar = world_map$region,
origin = "country.name",
destination = "iso2c", nomatch = NA)
data_cia <- data_cia %>% rename("Median age"=median_age,
                                "Youth unemployment rate"=youth_unempl_rate,
                                "Development status"=Developed...Developing.Countries,
                                "Region"=Region.Name,
                                "Subregion"=Sub.region.Name,
                                "Country"=country
                                )
final <- data_cia %>% right_join(world_map, by='iso2c')
display_df <- select(data_cia, !c("iso2c", "iso3c", "ISO3166.1.Alpha.3"))
```

Shiny App

NOTE: Please have patience while viewing the shiny app. The map visualization and changing between the variables takes some time to load.

Most problems I encountered were with the data table which would always overlap with the main panel graphs. I fixed this problem by using a data table proxy.

```
library(shiny)
library(plotly)
library(DT)

ui <-fluidPage(
  titlePanel("CIA World Factbook 2020"),
  sidebarLayout(
    sidebarPanel(selectInput("variable", "Select a variable:",
                    c("Youth unemployment rate"="`Youth unemployment rate`",
                      "Median age"="`Median age`",
                      "Population"), selected =c("Median age"="`Median age`")),
                 actionButton("button", "View raw data"),
                 DTOutput('tbl')
                 ),
    mainPanel(plotlyOutput("plot"))
  )
)

server <- function(input, output, session){
  observeEvent(input$button, {
    output$tbl <- renderDT(display_df,
                    options = list(scrollX = TRUE, paging=TRUE, pageLength=15))

    proxy <- dataTableProxy('tbl')
  })

  output$plot <- renderPlotly({
    plt <-
      ggplot(final, aes_string(x = "long", y = "lat",
                               group = "group", text="Country",
                               fill=input$variable)) +
            geom_polygon(colour='white') + scale_fill_viridis_c() + xlab("") + ylab("")
    plot <- ggplotly(plt, tooltip = c("text", "x", "y", "fill"))
  })
}
shinyApp(ui, server)
```