Data Visualisation and Communication (MATH2270)

Assignment 3: Storytelling with open data

Student Name: Vishwa Gandhi

Student Number: S3714805

URL to published assignment: https://vishwagandhi.shinyapps.io/TradePartners/

Final Code:

```
# Load required packages
library(plotly)
library(shiny)
library(shinydashboard)
library(tidyr)
library(ggplot2)
library(dplyr)
library(readxl)
library("ggsci")
library(extrafont)


# Dashboard UI design
ui <- dashboardPage(
  dashboardHeader(
    title = "Top Global Trade (Import/Export) Partners of Australia in 2018",
    titleWidth = "100%"
  ),
dashboardSidebar(
 disable = TRUE
),
dashboardBody(
 shinyUI(fluidPage(
  tags$head(tags$style(HTML("
              div.box-header {
               text-align: center;
              }
              ")))
 )),

 # Adding slider and select panel to fetch value of N and trade typr
 fluidRow(
  column(6,box( status = "primary",
         sliderInput("topn", "N: Number of top Trade Partners", min = 10, max = 30, value = 10),
         width = "100%"
  )),
  column(6,box( status = "primary",
         selectInput("tradetype", "Select a Trade Type",
               choices = c("Import" = "TotalImport", "Export" = "TotalExport")
         ),width = "100%")
  )),
 fluidRow(
  box( status = "primary", plotlyOutput("p1", height = 551)),
   box( status = "primary",title= "Select a country from Trade Partners to see Subsector Details",
     plotlyOutput("p2", height = 245)),
   box( status = "primary", plotlyOutput("p3", height = 245))
  ),
  fluidRow(
   uiOutput("srcref")
  )
 )
)

# Dashboard server code
server <-function(input, output, session) {
 # Loading all reuired data
 trade_data <- read_excel("import_export_data.xlsx", sheet = 2)
 import_sector <- read_excel("import_export_data.xlsx", sheet = 3)
 years_trade <- read_excel("import_export_data.xlsx", sheet = 1)

 # Setting general background color for dashboard
 background <- "#EFF1F0"

 # Generating First graph: Top N trade partners of Australia in 2018
 output$p1 <- renderPlotly({
  # preparing dataset for plot based on filter values selected
  import_val <<- trade_data %>% filter(trade_data$TradeType == input$tradetype)
  import_val <<- import_val[order(-import_val$Value),][0:input$topn,]
  import_val <<- import_val %>% mutate(valueK = round(Value/1000,3))

  import_val <<- import_val %>% arrange(valueK)


  # Horizontal barchart

  p1 <- ggplot(data = import_val, aes(x =reorder(Country, valueK), y =valueK, fill = valueK))

  p1 <- p1 + geom_bar(stat = "identity", width=0.8, aes( text = paste(valueK, "Thousand Dollars, ",
Country)))

  p1 <- p1 + labs(x = "Country", y = "Trade Value (thousand dollars)",

          title = paste("Top", input$topn, "Australian Rrade Partners of 2018"), fill = "Value")

  p1 <- p1 + theme_minimal()+ scale_fill_continuous(high="#581845", low="#FF5733")

  p1 <- p1 + theme(plot.background = element_rect(fill = background),

          panel.background = element_rect(fill = background),

          legend.background = element_rect(fill = background),

          plot.title = element_text(hjust = 0.5),

          text=element_text(family="Georgia"),

          title = element_text(face = "bold"),

          legend.title = element_text(face = "bold"))+ coord_flip()


  p1 <- ggplotly(p1, tooltip = "text", source = "p1")

  p1

 })


 # Generating Second Graph: Sector wise Trading for a Country

 output$p2 <- renderPlotly({

  # Fetch value of clicked event

  s_country <-  event_data("plotly_click", source = "p1")
```

```
if (length(s_country)){
 rank <- s_country[4]
 rank <- rank %>% as.numeric()
 print(rank)
 print(import_val)
 target_country <- import_val[rank,]$Country

 # Preparing data for the plot based on fetched value of trade type and country
 plot2_data <- import_sector %>% filter(Country == target_country, TradeType == input$tradetype)
 plot2_data <- plot2_data %>% mutate(valueK = round(ImportVal/1000, 3))

 # Bar chart
 p2 <- ggplot(data = plot2_data, aes(x = Sector, y =valueK))
 p2 <- p2 + geom_bar(stat = "identity", fill = "mediumaquamarine", width=0.8, aes(  text = paste(
  ImportVal, " Thousand Dollars, Sector:", Sector)))
 p2 <- p2 + labs(x = "Sector", y = "Trade Value (thousand dollars)",
         title = paste("Austrlalia's Sector wise", input$tradetype ," with ", target_country, " in 2018"))
 p2 <- p2 + theme_minimal()
 p2 <- p2 + theme(plot.background = element_rect(fill = background),
          panel.background = element_rect(fill = background),
          plot.title = element_text(hjust = 0.5),
          text=element_text(family="Georgia"),
          title = element_text(face = "bold"))

 p2 <- ggplotly(p2, tooltip = "text")
```

```
    p2
  }
  else{
    plotly_empty()
  }
})

# Preparing data for Trend analysis
years_trade %>% drop_na()
years_trade <- years_trade %>% mutate(AustralianTrade = round(austrade/1000, 3))

# Generating third Graph: Trade Trend of Australia
output$p3 <- renderPlotly({
  # Line graph for trend analysis
  p3 <- ggplot(data = years_trade, aes(x =Year, y = AustralianTrade,group = tradetype,
                                   color = tradetype, text = paste("Year:" , Year, ", ", tradetype, " Total Value:",
                                                                   AustralianTrade, " , Australia")))
  p3 <- p3 + geom_line()
  p3 <- p3 + labs(x = "Year", y = "Trade value (Thousand Dollars)",
                  title = "Trade Trend of Australia (2006-2018)")
  p3 <- p3 + theme_minimal() + scale_colour_brewer(palette = "Set1")
  p3 <- p3 + theme(plot.background = element_rect(fill = background),
                   panel.background = element_rect(fill = background),
                   legend.background = element_rect(fill = background),
                   plot.title = element_text(hjust = 0.5),
                    text=element_text(family="Georgia"),
                    #axis.text.x = element_text(angle = 45),
                    title = element_text(face = "bold"),
                    legend.title = element_text(face = "bold"),
                    legend.position = "right")
    p3 <- ggplotly(p3, tooltip = "text")
    p3
  })


  # Dataset reference
  output$srcref <- renderText(
    HTML(paste0("<b>", "Reference: ", "</b><i>", "Trade statistical pivot tables | DFAT. (2018). Retrieved June 17, 2020, from Dfat.gov.au website: https://www.dfat.gov.au/about-us/publications/Pages/trade-statistical-pivot-tables",
                "</i>")))


}


shinyApp(ui, server)
```