

Lab 3

Munachimso Aneke

2022-06-06

Explanation

My shiny app analyzes the Life Expectancy Data Set (<https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who> (<https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who>)). The data set used is a subset of the original data with the Year column converted to characters. The app uses tabs, checkboxes, numeric input boxes, and select input boxes (UI elements). The app contains three tabs: About, Analysis, and Table. The About tab contains information about the data set and the definition of the columns in the data subset. The Table tab displays the data set in table form. To control the number of observations and which columns to display, numeric input and checkboxes are employed. In the Analysis tab, the data is examined using simple linear regression. Users can control the data plotted on the y-axis and x-axis and a factor variable. The summary of the regression analysis is under the graph section.

Note

1. The CSV file has to be saved in the same directory and must match the name `Life Expectancy Data.csv`
2. The graph takes a little time to load.

Shiny App

```

library(shiny)
library(dplyr)
library(plotly)

# Load Data
life_df = read.csv("Life Expectancy Data.csv", sep = ",", header = TRUE)
life_df = na.omit(life_df)
life_df$Year = as.character(life_df$Year)
life_df = life_df %>% select(Country, Year, Status, Life.expectancy,
                             Adult.Mortality, infant.deaths, BMI,
                             Total.expenditure, Population,
                             Income.composition.of.resources)

# ui.R
ui <- fluidPage(
  titlePanel(h1(strong(("Life Expectancy Analysis")))),
  br(),
  tabsetPanel(
    tabPanel("About",
      p(h3(tags$u("Life Expectancy (WHO)")),
        "The data was collected from WHO and United Nations website with the help of D
eeksha Russell and Duan Wang.",
        "This app uses a subset of the original data. It contains 1649 observations and 1
0 columns with no missing values.",
        br(), br(),
        h4(tags$u("Column Definitions")),
        tags$ol(
          tags$li("Country: country name"),
          tags$li("Year: year"),
          tags$li("Status: developed or developing country"),
          tags$li("Life Expectancy: life expectancy in age"),
          tags$li("Adult Mortality: probability of dying between ages 15-60 per 1000 popu
lation"),
          tags$li("Infant Deaths: no of infant deaths per 1000 population"),
          tags$li("BMI: Average Body Mass Index of entire population"),
          tags$li("Total Expenditure: government expenditure on health as a percentage of
total government expenditure (%")),
          tags$li("Population: country's population"),
          tags$li("Income Composition: Human Development Index in terms of income composi
tion of resources (index ranging from 0 to 1)"),
        ),
        br(), br(),
      ),
    "Data Source: ",
    em(h6(a(href = 'https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who',
      "https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who")))),
    ),
    tabPanel("Analysis",
      sidebarLayout(
        sidebarPanel(
          selectInput("dependent", label = h4("Y axis"),
            choices = list("Life Expectancy" = "Life.expectancy",

```

```

        "Adult Mortality" = "Adult.Mortality",
        "Infant Deaths" = "infant.deaths",
        "BMI" = "BMI",
        "Total Expenditure" = "Total.expenditure",
        "Population" = "Population",
        "Income Composition" = "Income.composition.of.resources"),
    selected = 1),
  selectInput("independent", label = h4("X axis"),
    choices = list("Adult Mortality" = "Adult.Mortality",
      "Life Expectancy" = "Life.expectancy",
      "Infant Deaths" = "infant.deaths",
      "BMI" = "BMI",
      "Total Expenditure" = "Total.expenditure",
      "Population" = "Population",
      "Income Composition" = "Income.composition.of.resources"),
    selected = 1),
  selectInput("factor", label = h4("Factor"),
    choices = list("Country" = "Country",
      "Year" = "Year",
      "Status" = "Status"), selected = 1)
),
mainPanel(plotlyOutput("plot"),
  br(), br(),
  h4("Summary of Model"),
  verbatimTextOutput("summary"))
)),
tabPanel("Table",
  sidebarLayout(
    sidebarPanel(
      numericInput(inputId = "obs",
        label = "Number of observations to view:",
        value = 15),
      uiOutput("choose_columns")),
    mainPanel(tableOutput("table"))
  ))
)
)

```

server.R

```

server <- function(input, output){
  df <- reactive({life_df})

  # choose columns
  output$choose_columns <- renderUI({

    colnames <- names(df())
    # Create the check boxes and select 4 by default
    checkboxGroupInput("columns", "Choose columns to display",
      choices = colnames, selected = colnames[1:4])
  })
}

```

Display the table

```

output$table<- renderTable({

  # Keep the selected columns
  sel <- df()[, input$columns, drop = FALSE]
  # Return first n rows
  head(sel, n = input$obs)

})

# Display the plot
output$plot <- renderPlotly({
  model = lm(get(input$dependent)~get(input$independent), data = df())$fitted.values

  df_plot = df()%>% plot_ly(x = ~get(input$independent), y = ~get(input$dependent), color = ~get(input$factor), type = "scatter", mode = "markers") %>%
    layout( yaxis = list(title = input$dependent),
            xaxis = list(title = input$independent),
            legend=list(title=list(text = input$factor))) %>%
    add_trace(x = ~get(input$independent), y = model, mode = "lines", line = list(color = "red"))
})

# Display Summary
output$summary <- renderPrint({
  summary(lm(get(input$dependent)~get(input$independent), data = df()))
})
}

# Run App
shinyApp(ui = ui, server = server)

```

