Actividad N°5

Aplicacion de analisis de datos cuantitativos y cualitativos para encontrar la normalidad

Análisis Estadístico y Visualización Inteligente

Nombre:David Leon Callohuanca Condori Curso:Estadisitica Computacional Carrera:Ing. Estadisitica e Informatica Universidad:Universidad nacional del altiplano Fecha:21/04/2025

GITHUB

SHINY

FACEBOK

La aplicacion a realizar basicamente es basicamente encontrar la normalidad de forma rapida, interactiva y automatizada. Para eso la aplicacion estara hecha en shiny el cual es un framework para Rstudio. Permite al usuario:

- -Cargar sus propios datos en formato .csv.
- -Seleccionar una variable específica del conjunto de datos para su análisis.
- -Aplicar distintos tests de normalidad estadística:
- -Shapiro-Wilk
- -Kolmogorov-Smirnov
- -Lilliefors
- -Jarque-Bera
- -Generar automáticamente un reporte interpretativo usando la librería report, en caso de tratarse de una variable cuantitativa.

- -Visualizar los datos con diversos tipos de gráficos adaptativos, incluyendo:
- -Gráfico de barras
- -Gráfico circular (de sectores)
- -Gráfico de líneas
- -Diagrama de dispersión
- Histograma
- -Diagrama de caja y bigotes
- -Pictograma

Si el tipo de gráfico seleccionado no es compatible con el tipo de variable, se mostrará un aviso claro de "No compatible".

Sofware

El funcionamiento de codigo del programa se puede describir en varios bloques:

CARGA DE LIBRERIAS:

Permite importar las herramientas necesarias para el funcionamiento de la aplicación, incluyendo manejo de datos, generación de gráficos, realización de pruebas estadísticas y generación de reportes automáticos.

INTERFAZ DE USUARIO (UI):

Subida de archivo CSV: Permite al usuario cargar su base de datos con mas de 2 variables en formato .csv.

Selector dinámico de variable: Se genera un menú desplegable para elegir qué variable del archivo se va a analizar.

Selección del método estadístico: Ofrece los métodos Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors y Jarque-Bera para evaluar la normalidad de la variable.

Selección del tipo de gráfico: Una vez realizado el análisis, se habilita un menú para elegir el tipo de gráfico a generar (barras, circular, líneas, dispersión, histograma, caja y bigotes, pictograma).

BACKEND / SERVIDOR

Lectura del archivo CSV: Se procesa de manera reactiva el archivo subido por el usuario.

Actualización dinámica del selector de variables: Solo muestra las columnas del archivo cargado para que el usuario elija la que desea analizar.

Prueba de normalidad: Ejecuta la prueba seleccionada sobre la variable elegida, si esta es numérica. Muestra los resultados en pantalla.

Generación del gráfico: En función del tipo de variable (cuantitativa o cualitativa) y del tipo de gráfico seleccionado, se genera el gráfico correspondiente. Si no es compatible, se muestra un mensaje indicándolo.

Reporte automático: Utiliza la librería report para generar un resumen estadístico e interpretativo de la variable numérica seleccionada.

LOGICA CONDICIONAL

Condicional en análisis y visualización: El selector de gráficos y la sección de resultados solo se activan tras presionar el botón "Ejecutar análisis".

Compatibilidad de gráficos: La app evalúa si el gráfico seleccionado es compatible con el tipo de variable y actúa en consecuencia.

EJECUCION FINAL

Inicialización de la app: Llama a la función shinyApp(ui, server) para iniciar la aplicación combinando interfaz y servidor.

BLOQUES DE CODIGO

```
#CARGA DE LIBRERIAS:
library(shiny)
library(ggplot2)
library(dplyr)
library(nortest)
library(tseries)
library(moments)
library(report)
library(report)
library(readr)
#INTERFAZ DE USUARIO (UI):
ui <- fluidPage(
   titlePanel("Análisis Estadístico y Visualización Inteligente"),
   sidebarLayout(
    sidebarPanel(
        fileInput("datafile", "Sube tu archivo CSV", accept = ".csv"),</pre>
```

```
uiOutput("varselect"),
      selectInput("metodo", "Método estadístico",
                  choices = c("Shapiro-Wilk", "Kolmogorov-Smirnov", "Lilliefor
                  s", "Jarque-Bera")),
      actionButton("analizar", "Ejecutar análisis"),
      conditionalPanel(
        condition = "input.analizar > 0",
        selectInput("grafico", "Tipo de gráfico",
                    choices = c("Barras", "Circular", "Líneas", "Dispersión",
                                 "Histograma", "Caja y bigotes", "Pictograma"))
      )
    ),
    mainPanel(
      conditionalPanel(
        condition = "input.analizar > 0",
        h4("Resultado del análisis"),
        verbatimTextOutput("resultado"),
        h4("Gráfico seleccionado"),
        uiOutput("graficoUI"),
        h4("Reporte automático"),
        verbatimTextOutput("reporte")
   )
 )
server <- function(input, output, session) {</pre>
 datos <- reactive({</pre>
   req(input$datafile)
   read_csv(input$datafile$datapath)
 })
 output$varselect <- renderUI({</pre>
   req(datos())
    selectInput("variable", "Selecciona una variable", choices = names(datos()
                                                                          ))
 })
#BACKEND / SERVIDOR
  output$resultado <- renderPrint({</pre>
```

```
req(input$analizar)
    isolate({
      df <- datos()</pre>
      var <- input$variable</pre>
      x <- na.omit(df[[var]])</pre>
      if (!is.numeric(x)) {
        cat("La variable seleccionada no es numérica. El test no se puede apli
           car.")
      } else {
        switch(input$metodo,
               "Shapiro-Wilk" = print(shapiro.test(x)),
                "Kolmogorov-Smirnov" = print(ks.test(x, "pnorm", mean(x), sd(x)
                )),
                "Lilliefors" = print(lillie.test(x)),
                "Jarque-Bera" = print(jarque.bera.test(x))
        )
      }
    })
 })
#LOGICA CONDICIONAL
  output$graficoUI <- renderUI({</pre>
   req(input$grafico)
   plotOutput("grafico")
 })
 output$grafico <- renderPlot({</pre>
    req(input$analizar, input$grafico)
    isolate({
      df <- datos()</pre>
      var <- input$variable</pre>
      x <- df[[var]]
      tipo <- ifelse(is.numeric(x), "cuantitativa", "cualitativa")</pre>
      if (input$grafico == "Histograma" && tipo == "cuantitativa") {
        ggplot(df, aes(x = .data[[var]])) +
          geom_histogram(fill = "skyblue", bins = 30, color = "black") +
          theme_minimal()
      } else if (input$grafico == "Caja y bigotes" && tipo == "cuantitativa")
```

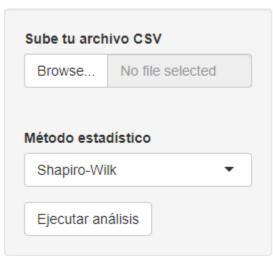
```
ggplot(df, aes(y = .data[[var]])) +
          geom_boxplot(fill = "orange") +
          theme_minimal()
      } else if (input$grafico == "Líneas" && tipo == "cuantitativa") {
        ggplot(df, aes(x = seq_along(.data[[var]]), y = .data[[var]])) +
          geom_line(color = "blue") +
          theme_minimal()
      } else if (input$grafico == "Dispersión" && tipo == "cuantitativa") {
        ggplot(df, aes(x = seq_along(.data[[var]]), y = .data[[var]])) +
          geom_point(color = "darkred") +
          theme_minimal()
      } else if (input$grafico == "Barras" && tipo == "cualitativa") {
        ggplot(df, aes(x = .data[[var]])) +
          geom_bar(fill = "steelblue") +
          theme_minimal()
      } else if (input$grafico == "Circular" && tipo == "cualitativa") {
        ggplot(df, aes(x = "", fill = .data[[var]])) +
          geom bar(width = 1) +
          coord_polar("y") +
          theme_void()
      } else if (input$grafico == "Pictograma" && tipo == "cualitativa") {
        ggplot(df, aes(x = .data[[var]])) +
          geom_point(stat = "count", size = 10, shape = 21, fill = "purple") +
          theme_minimal()
     } else {
        ggplot() +
          annotate("text", x = 0.5, y = 0.5, label = "No compatible", size =
          8, color = "red") +
         theme_void()
    })
 })
#EJECUCION FINAL
  output$reporte <- renderPrint({</pre>
```

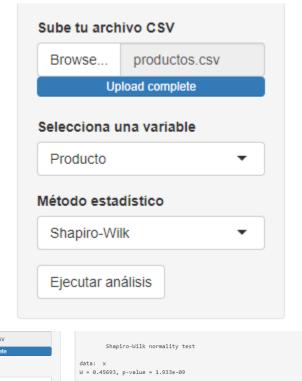
```
req(input$analizar)
isolate({
    df <- datos()
    var <- input$variable
    x <- na.omit(df[[var]])

    if (is.numeric(x)) {
        print(report(x))
    } else {
        cat("Solo se generan reportes para variables numéricas.")
    }
    })
}
shinyApp(ui, server)</pre>
```

Capturas

Análisis Estadístico y Visualización Inteligente







Reporte automático

x: n = 30, Mean = 100.60, SD = 163.60, Median = 52.50, MAD = 34.84, range: [19.99, 899.99], SI

