

Universidad Nacional del Altiplano

Facultad de Ingeniería Estadística e Informática

Docente: Fred Torres Cruz

Autor: Eva Ruth Mamani Josec

Repositorio de GitHub: https://github.com/evaruth270/app_est_comp01.git

Trabajo Encargado - N° 002

APLICACIÓN SHINY DE PRUEBAS ESTADÍSTICAS

Introducción

Esta aplicación web desarrollada en R usando el paquete Shiny tiene como propósito facilitar la elección de la prueba estadística más adecuada para comparar grupos en un conjunto de datos.

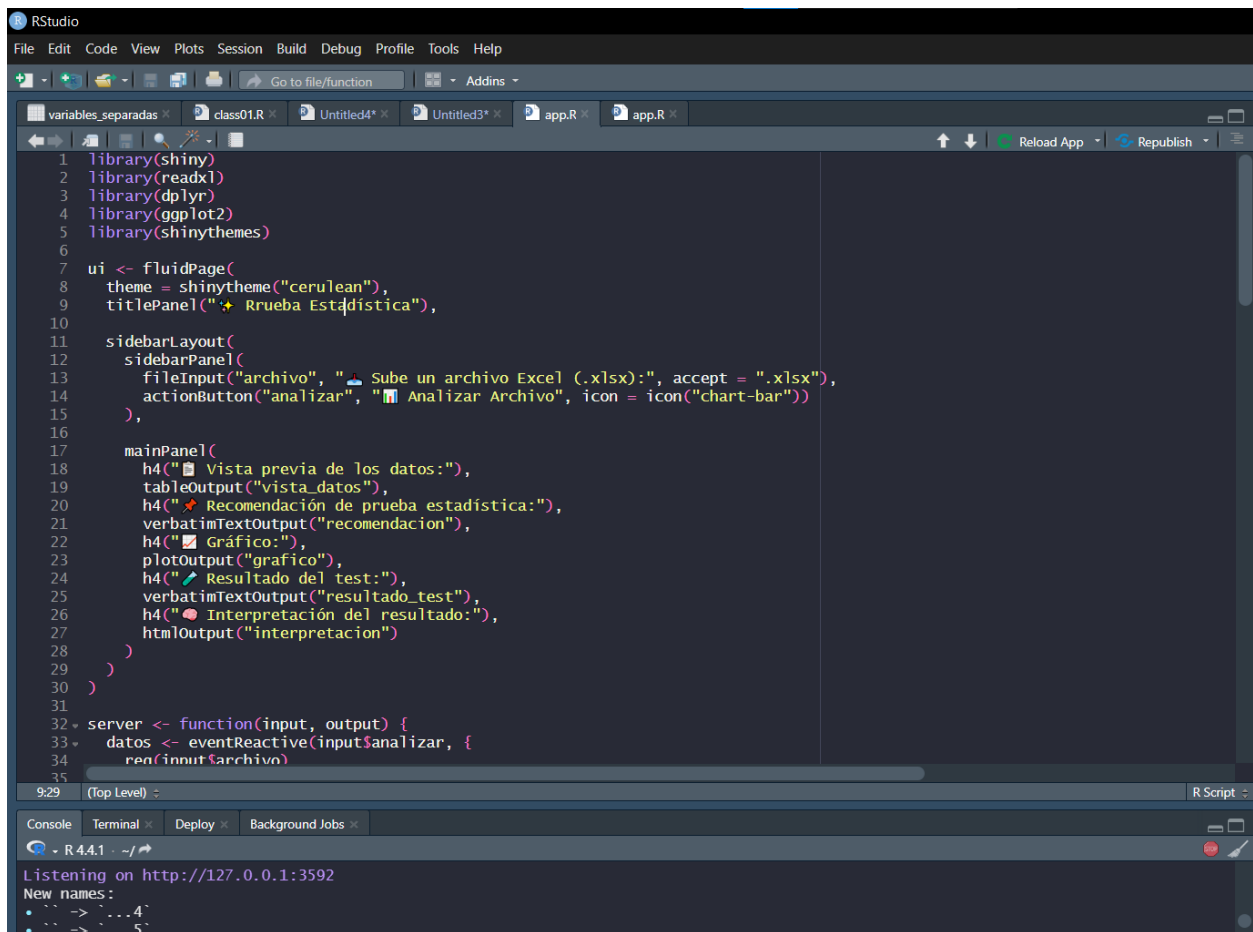
El funcionamiento de la aplicación es el siguiente:

- El usuario puede subir un archivo en formato Excel (.xlsx) que contenga datos categóricos (como género, grupo, curso, etc.) y datos numéricos (como puntajes, horas de estudio, etc.).
- La aplicación analiza automáticamente la variable categórica seleccionada y detecta cuántos grupos diferentes existen.
- Si hay exactamente dos grupos, la aplicación recomienda realizar una prueba t de Student.
- Si hay más de dos grupos, se recomienda realizar un análisis de varianza (ANOVA).
- Adicionalmente, se genera un gráfico tipo boxplot que permite visualizar la comparación entre los grupos.
- Finalmente, se ejecuta automáticamente la prueba estadística recomendada y se muestran sus resultados, incluyendo valores como el estadístico t o F, y el p-valor.

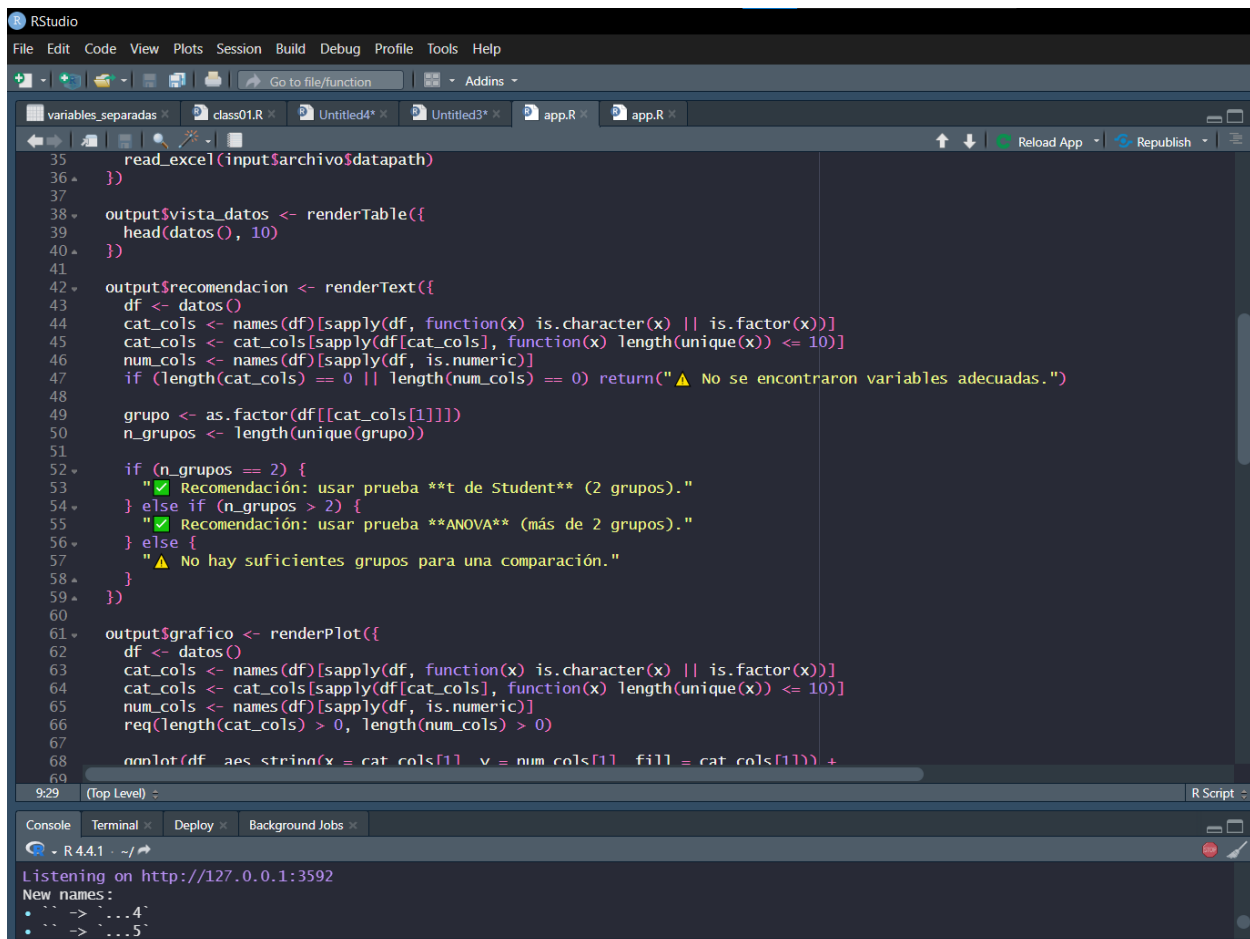
Esta herramienta está pensada para estudiantes o investigadores que necesitan una guía rápida para saber qué tipo de prueba utilizar para comparar medias entre grupos de datos, sin necesidad de escribir mucho código.

A continuación, se mostrarán capturas de pantalla que ilustran el funcionamiento de la aplicación.

1. Captura del código en RStudio



2. Código subido al repositorio en GitHub



The screenshot shows the RStudio interface with a script editor containing R code. The code defines a function `read_excel` that reads an Excel file, processes the data, and generates a report with a table and a plot. The console at the bottom shows the R version and a message about listening on a specific port.

```
35 read_excel(input$archivo$datapath)
36 })
37
38 output$vista_datos <- renderTable({
39   head(datos(), 10)
40 })
41
42 output$recomendacion <- renderText({
43   df <- datos()
44   cat_cols <- names(df)[sapply(df, function(x) is.character(x) || is.factor(x))]
45   cat_cols <- cat_cols[sapply(df[cat_cols], function(x) length(unique(x)) <= 10)]
46   num_cols <- names(df)[sapply(df, is.numeric)]
47   if (length(cat_cols) == 0 || length(num_cols) == 0) return("⚠ No se encontraron variables adecuadas.")
48
49   grupo <- as.factor(df[[cat_cols[1]]])
50   n_grupos <- length(unique(grupo))
51
52   if (n_grupos == 2) {
53     "✅ Recomendación: usar prueba **t de Student** (2 grupos)."
54   } else if (n_grupos > 2) {
55     "✅ Recomendación: usar prueba **ANOVA** (más de 2 grupos)."
56   } else {
57     "⚠ No hay suficientes grupos para una comparación."
58   }
59 })
60
61 output$grafico <- renderPlot({
62   df <- datos()
63   cat_cols <- names(df)[sapply(df, function(x) is.character(x) || is.factor(x))]
64   cat_cols <- cat_cols[sapply(df[cat_cols], function(x) length(unique(x)) <= 10)]
65   num_cols <- names(df)[sapply(df, is.numeric)]
66   req(length(cat_cols) > 0, length(num_cols) > 0)
67
68   ggplot(df, aes(string(x = cat_cols[1], y = num_cols[1], fill = cat_cols[1])) +
69
```

9:29 (Top Level) R Script

Console Terminal Deploy Background Jobs

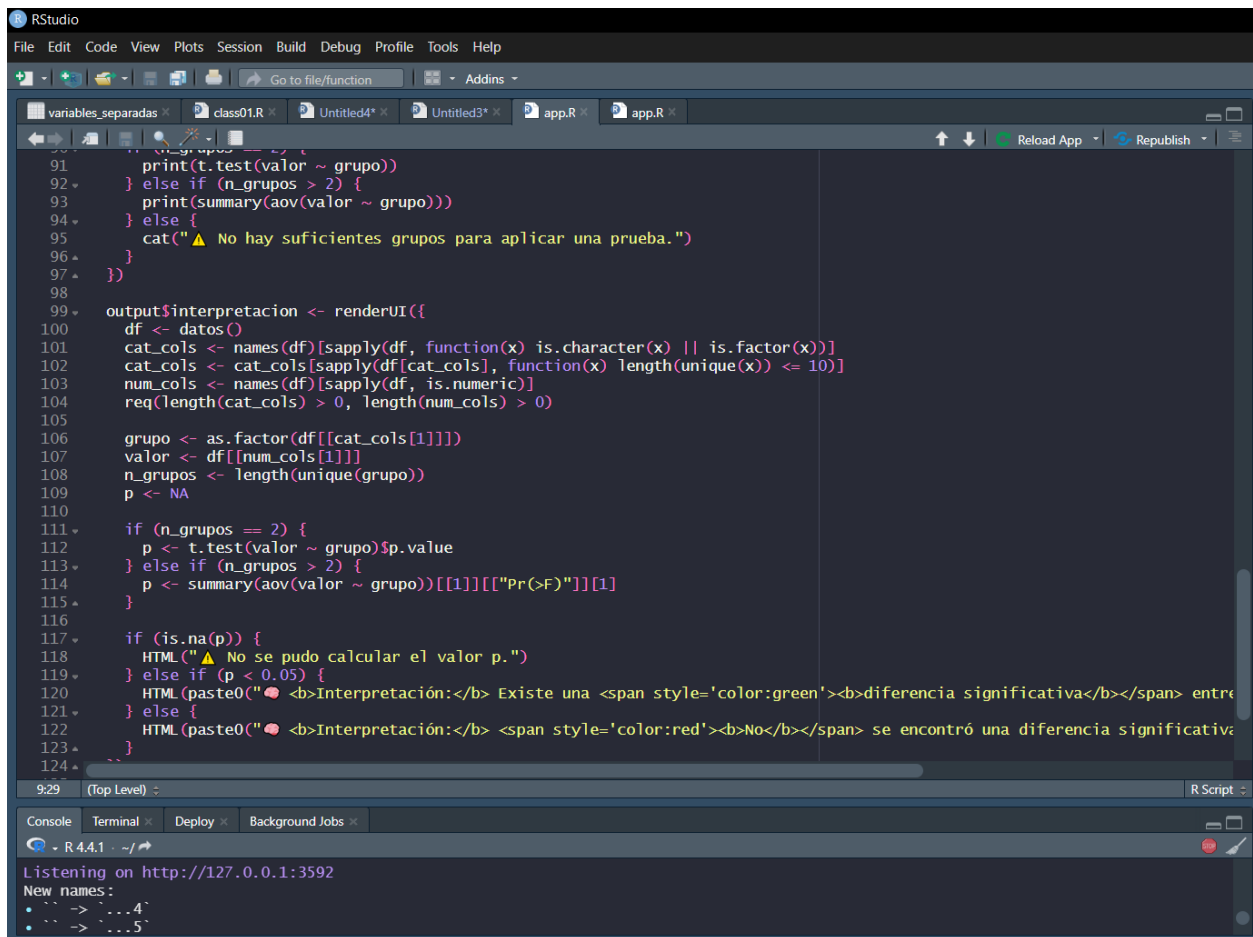
R 4.4.1 ~ /

Listening on http://127.0.0.1:3592

New names:

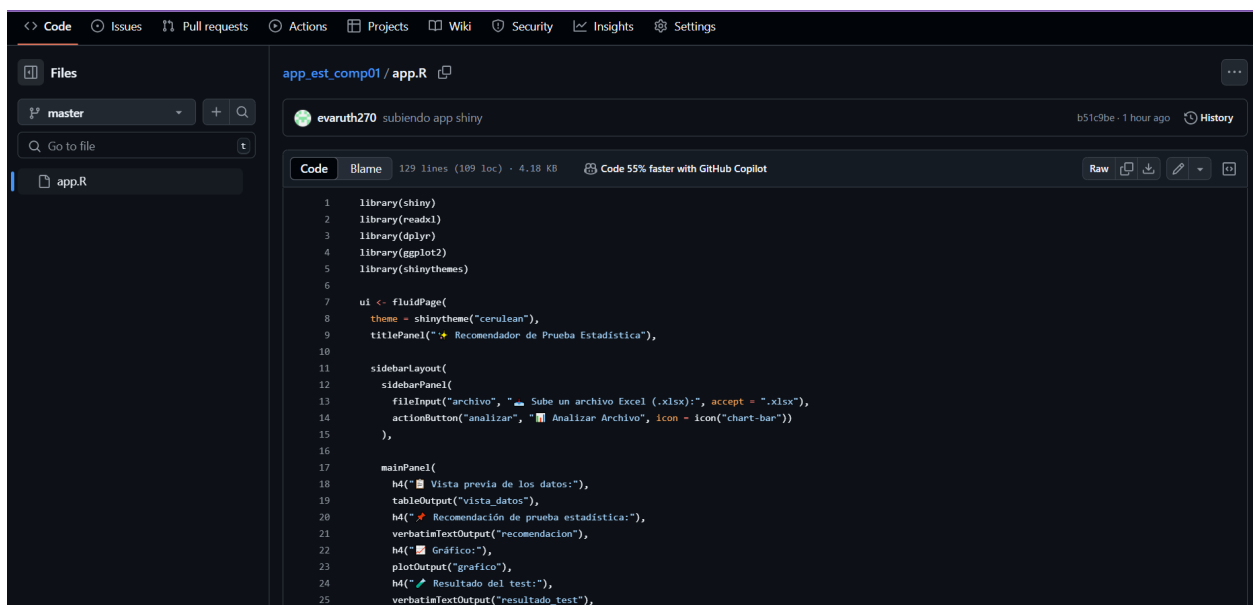
- -> ...4
- -> ...5

3. Aplicación en Shiny



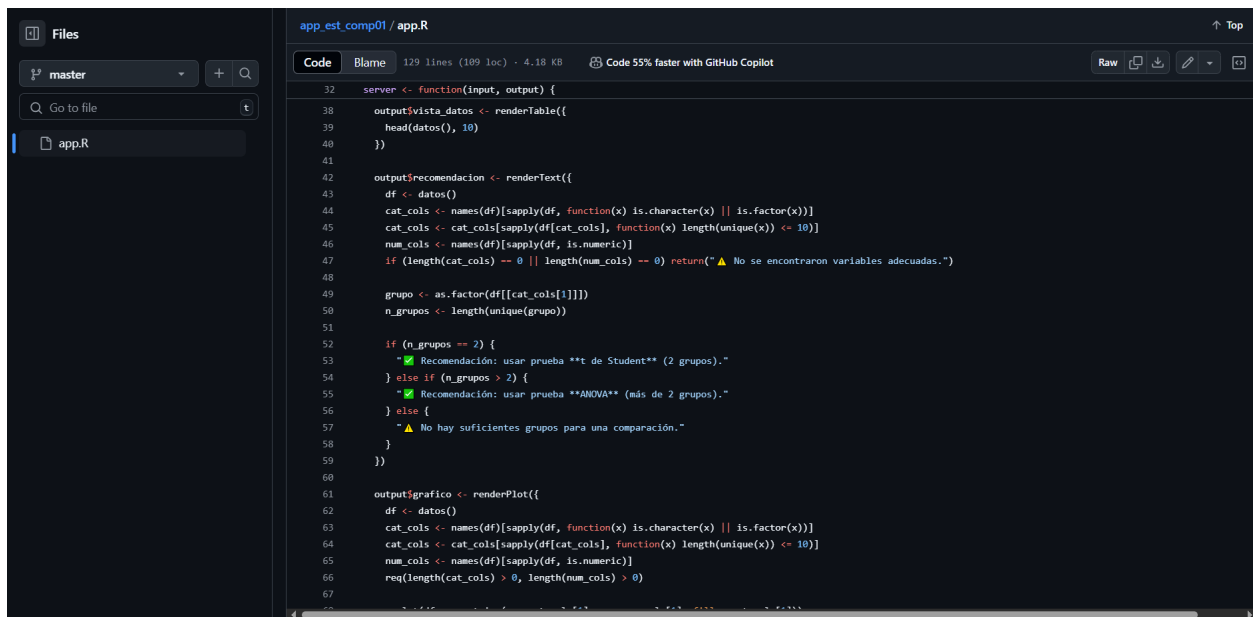
```
90 print(t.test(valor ~ grupo))
91 } else if (n_grupos > 2) {
92   print(summary(aov(valor ~ grupo)))
93 } else {
94   cat("⚠ No hay suficientes grupos para aplicar una prueba.")
95 }
96 }
97 })
98
99 output$interpretacion <- renderUI({
100   df <- datos()
101   cat_cols <- names(df)[sapply(df, function(x) is.character(x) || is.factor(x))]
102   cat_cols <- cat_cols[sapply(df[cat_cols], function(x) length(unique(x)) <= 10)]
103   num_cols <- names(df)[sapply(df, is.numeric)]
104   req(length(cat_cols) > 0, length(num_cols) > 0)
105
106   grupo <- as.factor(df[[cat_cols[1]])
107   valor <- df[[num_cols[1]]]
108   n_grupos <- length(unique(grupo))
109   p <- NA
110
111   if (n_grupos == 2) {
112     p <- t.test(valor ~ grupo)$p.value
113   } else if (n_grupos > 2) {
114     p <- summary(aov(valor ~ grupo))[[1]][["Pr(>F)"]][1]
115   }
116
117   if (is.na(p)) {
118     HTML("⚠ No se pudo calcular el valor p.")
119   } else if (p < 0.05) {
120     HTML(paste0("🟢 <b>Interpretación:</b> Existe una <span style='color:green'><b>diferencia significativa</b></span> entre")
121   } else {
122     HTML(paste0("🔴 <b>Interpretación:</b> <span style='color:red'><b>No</b></span> se encontró una diferencia significativa")
123   }
124 })
```

4. Aplicación en el navegador



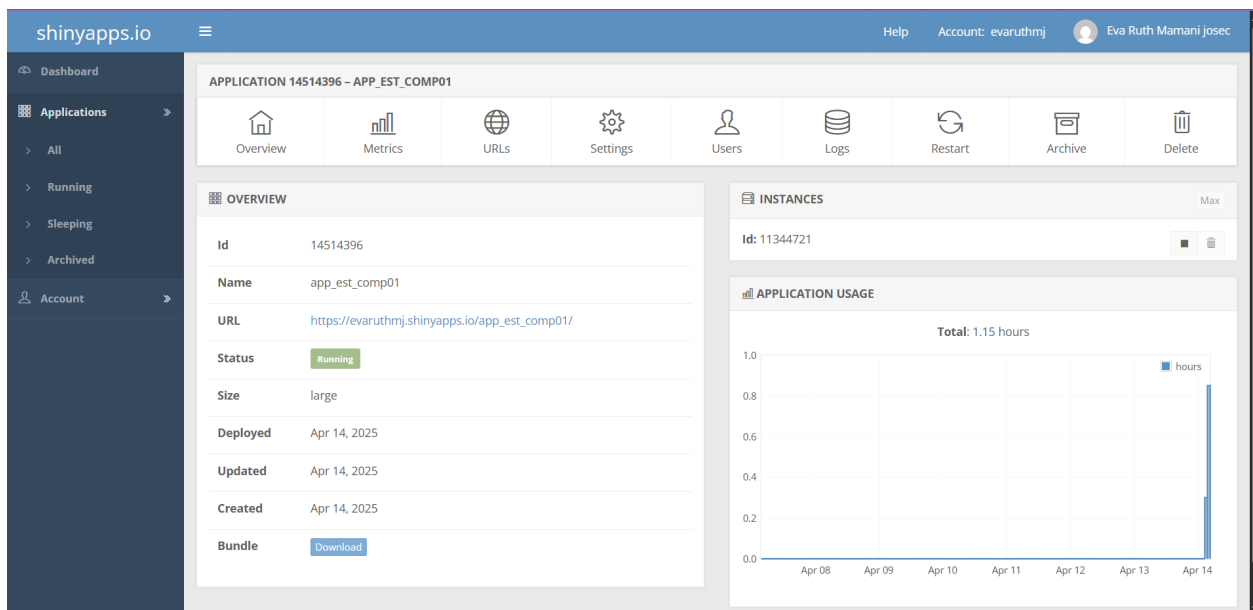
```
1 library(shiny)
2 library(readxl)
3 library(dplyr)
4 library(ggplot2)
5 library(shinythemes)
6
7 ui <- fluidPage(
8   theme = shinytheme("cerulean"),
9   titlePanel("🌟 Recomendador de Prueba Estadística"),
10
11   sidebarLayout(
12     sidebarPanel(
13       fileInput("archivo", "📎 Sube un archivo Excel (.xlsx):", accept = ".xlsx"),
14       actionButton("analizar", "📊 Analizar Archivo", icon = icon("chart-bar"))
15     ),
16
17     mainPanel(
18       h4("📄 Vista previa de los datos:"),
19       tableOutput("vista_datos"),
20       h4("🌟 Recomendación de prueba estadística:"),
21       verbatimTextOutput("recomendacion"),
22       h4("📊 Gráfico:"),
23       plotOutput("grafico"),
24       h4("📈 Resultado del test:"),
25       verbatimTextOutput("resultado_test"),
```

5. Gráfico boxplot generado



```
32 server <- function(input, output) {
33
34   output$vista_datos <- renderTable({
35     head(datos(), 10)
36   })
37
38   output$recomendacion <- renderText({
39     df <- datos()
40     cat_cols <- names(df)[sapply(df, function(x) is.character(x) || is.factor(x))]
41     cat_cols <- cat_cols[sapply(df[cat_cols], function(x) length(unique(x)) <= 10)]
42     num_cols <- names(df)[sapply(df, is.numeric)]
43     if (length(cat_cols) == 0 || length(num_cols) == 0) return("⚠ No se encontraron variables adecuadas.")
44
45     grupo <- as.factor(df[[cat_cols[1]]])
46     n_grupos <- length(unique(grupo))
47
48     if (n_grupos == 2) {
49       "✅ Recomendación: usar prueba **t de Student** (2 grupos)."
```

6. Resultados de la prueba t o ANOVA



7. Subida del archivo Excel

← → ↻ evaruthnj.shinyapps.io/app_est_comp01/ ☆ ⬇ 📄 🌐

Rrueba Estadística

Sube un archivo Excel (.xlsx):

Browse... No file selected

Analizar Archivo

Vista previa de los datos:

Recomendación de prueba estadística:

Gráfico:

Resultado del test:

Interpretación del resultado:

8. Menú de selección de variables

← → ↻ evaruthnj.shinyapps.io/app_est_comp01/ ☆ ⬇ 📄 🌐

Rrueba Estadística

Sube un archivo Excel (.xlsx):

Browse... Ejercicio Tamaño de muestra.xlsx

Upload complete

Analizar Archivo

Vista previa de los datos:

Orden	Edad	Vivienda_Propia	...4	...5	...6	...7	...8	...9	...10	...11	...12
1.00	34.00	SI	670.00	no	821.00	NA	1	46.00	SI	325.00	si
2.00	53.00	SI	176.00	si	497.00	NA	2	30.00	SI	54.00	si
3.00	32.00	SI	987.00	no	783.00	NA	3	38.00	SI	158.00	si
4.00	49.00	SI	389.00	si	681.00	NA	4	30.00	SI	54.00	si
5.00	30.00	SI	18.00	si	72.00	NA	5	38.00	SI	158.00	si
6.00	30.00	SI	109.00	si	42.00	NA	6	64.00	SI	91.00	si
7.00	19.00	No	756.00	no	55.00	NA	7	21.00	No	139.00	si
8.00	40.00	No	87.00	si	690.00	NA	8	39.00	No	359.00	si
9.00	60.00	No	14.00	si	932.00	NA	9	31.00	SI	196.00	si
10.00	24.00	No	486.00	no	822.00	NA	10	64.00	SI	323.00	si

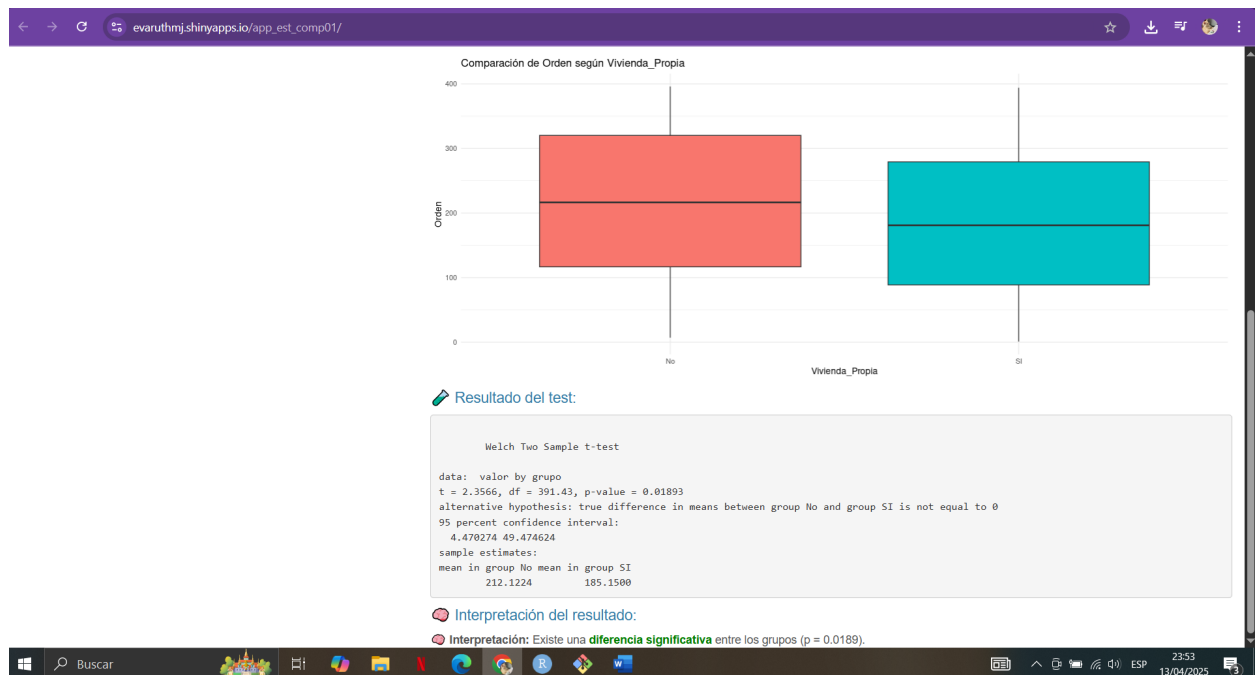
Recomendación de prueba estadística:

Recomendación: usar prueba **t de Student** (2 grupos).

Gráfico:

Comparación de Orden según Vivienda_Propia

9. Vista general de la aplicación



Conclusión

La aplicación desarrollada permite cargar un archivo Excel y, de forma automática, recomendar la prueba estadística adecuada (t de Student o ANOVA) según el número de grupos de la variable categórica. Es una herramienta útil, práctica y fácil de usar para comparar grupos de forma rápida y visual.