

Análisis Espacial con Datos Geográficos

Matrices de Pesos, Índice de Moran, Índice de Geary y Hotspots

Autor: Katia Daishy Ticona Casa

Curso: Estadística Espacial

Año: 2025

Introducción

El análisis espacial permite estudiar cómo las variables observadas en distintas ubicaciones geográficas se relacionan entre sí. A través de herramientas estadísticas especializadas, es posible identificar patrones, dependencias y concentraciones espaciales (clústeres), esenciales en campos como la geografía, economía, sociología o agronomía.

En este documento se describen los principales conceptos empleados en el análisis espacial implementado en **R**: las *matrices de pesos espaciales*, los *índices de autocorrelación espacial* (Moran y Geary) y el análisis de *Hotspots* (*Getis-Ord G_i^**).

1. Matrices de Pesos Espaciales

La **matriz de pesos espaciales** (W) es la base del análisis espacial. Representa las relaciones de vecindad entre las unidades geográficas. Cada elemento w_{ij} indica el grado de conexión o influencia entre las observaciones i y j .

Definición

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}$$

Donde:

- $w_{ij} = 1$ si las unidades i y j son vecinas.
- $w_{ij} = 0$ en caso contrario.

Los pesos pueden definirse según:

1. **Contigüidad:** los polígonos comparten frontera.

2. **Distancia:** puntos dentro de un radio específico.
3. **Vecinos más cercanos:** por ejemplo, $k = 5$ vecinos más próximos.

En R, se construyen con:

```
nb <- knn2nb(knearneigh(coords, k = 5))  
lw <- nb2listw(nb, style = "W")
```

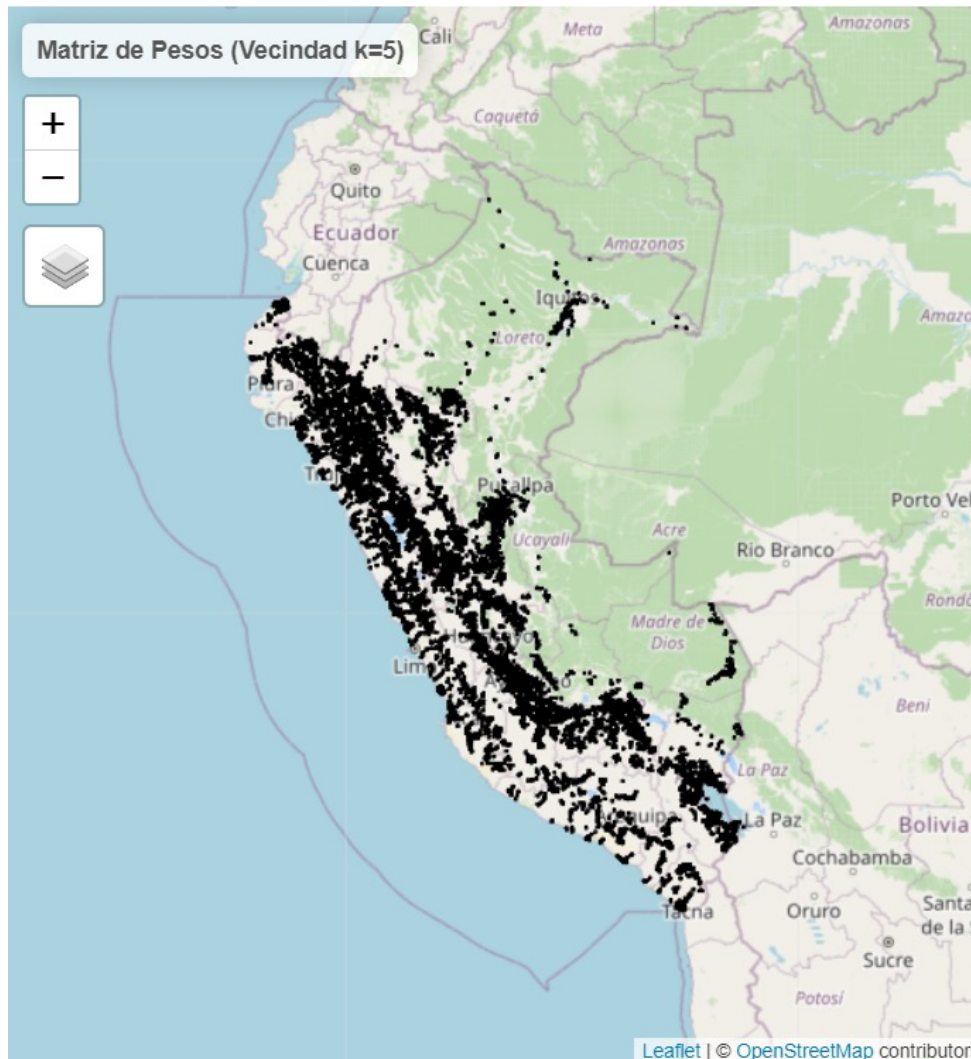


Figura 1: Mapa de Matriz de Pesos

La matriz de pesos espaciales define las relaciones de vecindad entre las unidades geográficas del estudio. En este caso, se utilizó el método de los **5 vecinos más cercanos** ($k = 5$), donde cada punto está conectado con sus cinco observaciones más próximas según su posición geográfica (latitud y longitud).

En el mapa se observa la distribución de las unidades principalmente en la zona costera y andina del Perú, con menor densidad en la Amazonía. Esto indica que la estructura espacial se concentra en regiones con mayor actividad productiva y densidad poblacional.

Esta matriz constituye la base para los siguientes análisis espaciales: el *Índice de Moran*, el *Índice de Geary* y la detección de *Hotspots*, los cuales dependen de la correcta definición de la vecindad. En resumen, el modelo $k=5$ permite representar adecuadamente la dependencia espacial en el territorio de estudio.

2. Índice de Moran

El **Índice de Moran** (I) mide la autocorrelación espacial global, es decir, si valores similares se agrupan en el espacio o se distribuyen aleatoriamente.

Fórmula

$$I = \frac{n}{S_0} \cdot \frac{\sum_i \sum_j w_{ij} (x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2}$$

Donde:

- n : número de observaciones.
- x_i : valor de la variable en la unidad i .
- \bar{x} : media de la variable.
- w_{ij} : elemento de la matriz de pesos espaciales.
- $S_0 = \sum_i \sum_j w_{ij}$: suma total de los pesos.

Interpretación

- $I > 0$: autocorrelación positiva (valores similares tienden a agruparse).
- $I < 0$: autocorrelación negativa (valores altos rodeados de bajos y viceversa).
- $I \approx 0$: distribución aleatoria.

Ejemplo en R

```
moran.test(sf_data$P104_SUP_ha, lw)
```

Este mapa representa la distribución espacial de una variable categorizada como var-clean en el territorio peruano y zonas limítrofes. El valor del Índice de Moran (0.0713) indica una autocorrelación espacial muy débil, lo que sugiere que los valores similares no tienden a agruparse de forma significativa en el espacio.

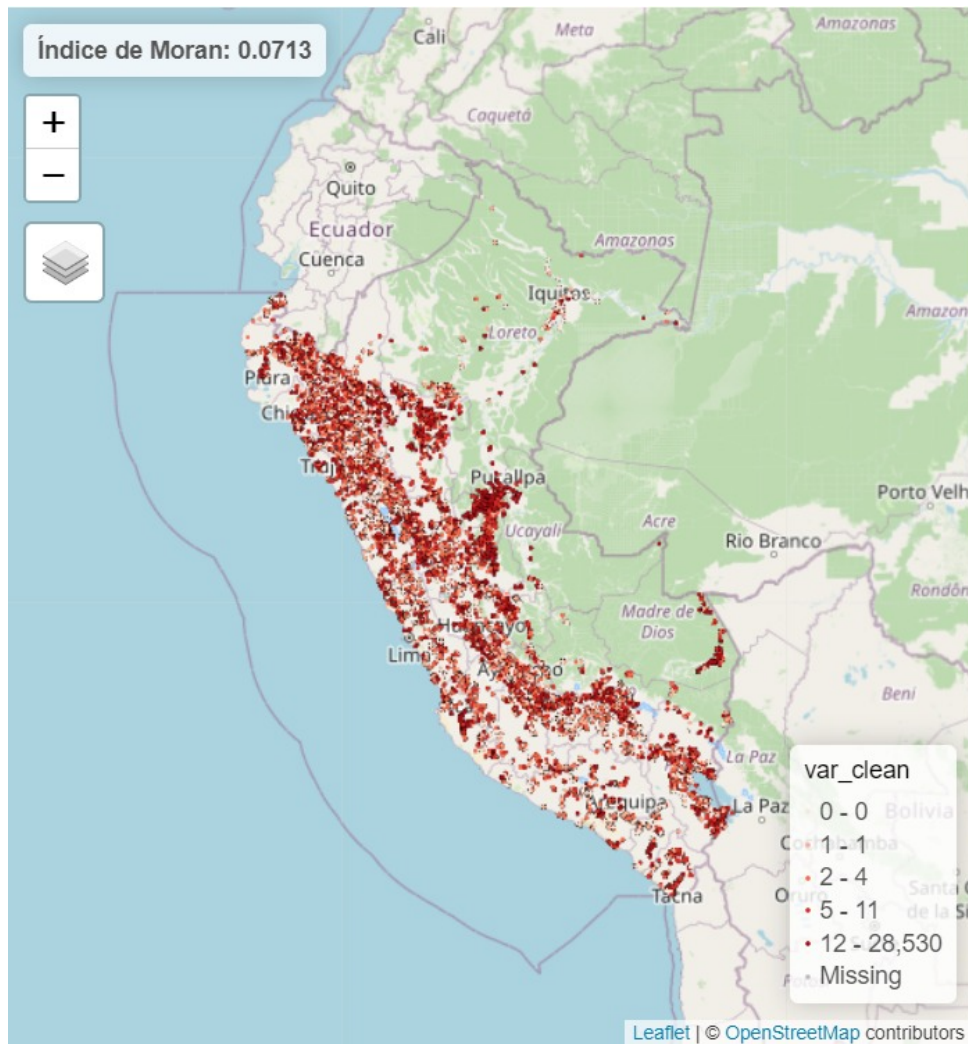


Figura 2: Mapa de Índice de Moran

Resultados del test de Moran I

- Estadístico Moran I: 0,0713
- Valor esperado bajo aleatorización: $-0,00003$
- Varianza: 0,00000773
- Desviación estándar del estadístico: 25,657
- Valor-p: $< 2,2 \times 10^{-16}$
- Hipótesis alternativa: La autocorrelación espacial es mayor que la esperada bajo aleatorización.

El valor observado del Índice de Moran I es 0.0713, ligeramente positivo, lo que indica una autocorrelación espacial débil

Sin embargo, el valor-p extremadamente bajo (2.2×10^{-1}) sugiere que esta autocorrelación, aunque débil, es estadísticamente significativa

3. Índice de Geary

El **Índice de Geary** (C) también mide autocorrelación espacial, pero es más sensible a las diferencias locales entre vecinos.

Fórmula

$$C = \frac{(n-1)}{2S_0} \cdot \frac{\sum_i \sum_j w_{ij} (x_i - x_j)^2}{\sum_i (x_i - \bar{x})^2}$$

Interpretación

- $C < 1$: autocorrelación positiva.
- $C = 1$: distribución aleatoria.
- $C > 1$: autocorrelación negativa.

Ejemplo en R

```
geary.test(sf_data$P104_SUP_ha, lw)
```

El valor del Índice de Geary (0.8334) indica una autocorrelación espacial débil, lo que sugiere que los valores de la variable no presentan una fuerte similitud entre unidades vecinas; es decir, hay una distribución relativamente aleatoria.

Se observa una mayor concentración de datos en la región occidental y central del país, especialmente en la zona andina y costera, mientras que la región amazónica oriental presenta menor densidad de observaciones.

Resultados del test de Geary C

- **Estadístico Geary C:** 0,8334
- **Valor esperado bajo aleatorización:** 1,0000
- **Varianza:** 0,01596
- **Desviación estándar del estadístico:** 1,3183

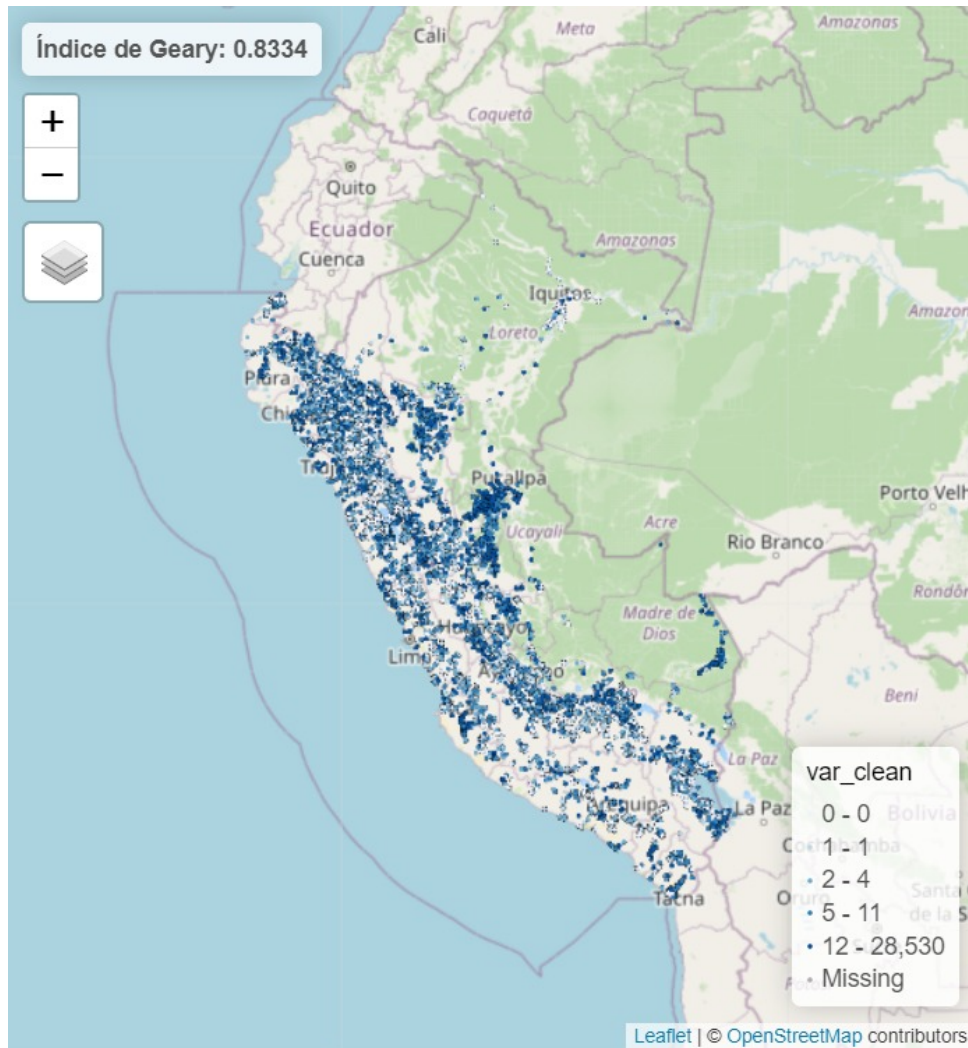


Figura 3: Mapa de Índice de Geary

- **Valor-p:** 0,09371
- **Hipótesis alternativa:** La expectativa es mayor que el estadístico observado.

El valor observado del Índice de Geary C es 0.8334, inferior al valor esperado bajo aleatorización (1.0000), lo que sugiere una autocorrelación espacial positiva leve

Sin embargo, el valor-p de 0.09371 indica que esta autocorrelación no es estadísticamente significativa al nivel convencional de 0.05.

4. Hotspots (Getis-Ord G_i^*)

El estadístico **Getis-Ord G_i^*** identifica clústeres espaciales de valores altos (*hot spots*) o bajos (*cold spots*). Es un análisis *local* de autocorrelación espacial.

Fórmula

$$G_i^* = \frac{\sum_j w_{ij}x_j - \bar{X} \sum_j w_{ij}}{S \sqrt{\frac{[n \sum_j w_{ij}^2 - (\sum_j w_{ij})^2]}{n-1}}}$$

Donde:

- x_j : valor observado en la unidad j .
- \bar{X} : media global de la variable.
- S : desviación estándar global.

Interpretación

- G_i^* alto y significativo \Rightarrow **Hotspot** (concentración de valores altos).
- G_i^* bajo y significativo \Rightarrow **Coldspot** (concentración de valores bajos).

Ejemplo en R

```
gi <- localG(sf_data$P104_SUP_ha, lw)
sf_data$Gi <- as.numeric(gi)
```

El análisis de *Hotspots* mediante el estadístico de Getis-Ord G_i^* permite identificar zonas donde los valores altos o bajos de una variable se agrupan de manera significativa. Cada punto del mapa representa una unidad espacial, y el color refleja el nivel de intensidad o concentración local del fenómeno.

Los tonos **rojos y naranjas** indican las áreas denominadas **Hotspots**, es decir, zonas con valores altos que tienden a agruparse espacialmente. Estas concentraciones se observan principalmente en la **costa y sierra sur del Perú**, destacando regiones como Arequipa, Cusco, Puno y Lima, donde el fenómeno presenta mayor intensidad. Por otro lado, los tonos **amarillos o claros** corresponden a valores medios o sin significancia estadística, y las áreas sin color representan zonas sin datos o con valores no significativos.

Conclusiones

El análisis espacial proporciona herramientas potentes para descubrir patrones no evidentes en datos geográficos. Mientras el índice de **Moran** detecta autocorrelación global, el índice de **Geary** permite examinar variaciones locales y el estadístico **Getis-Ord G_i^***

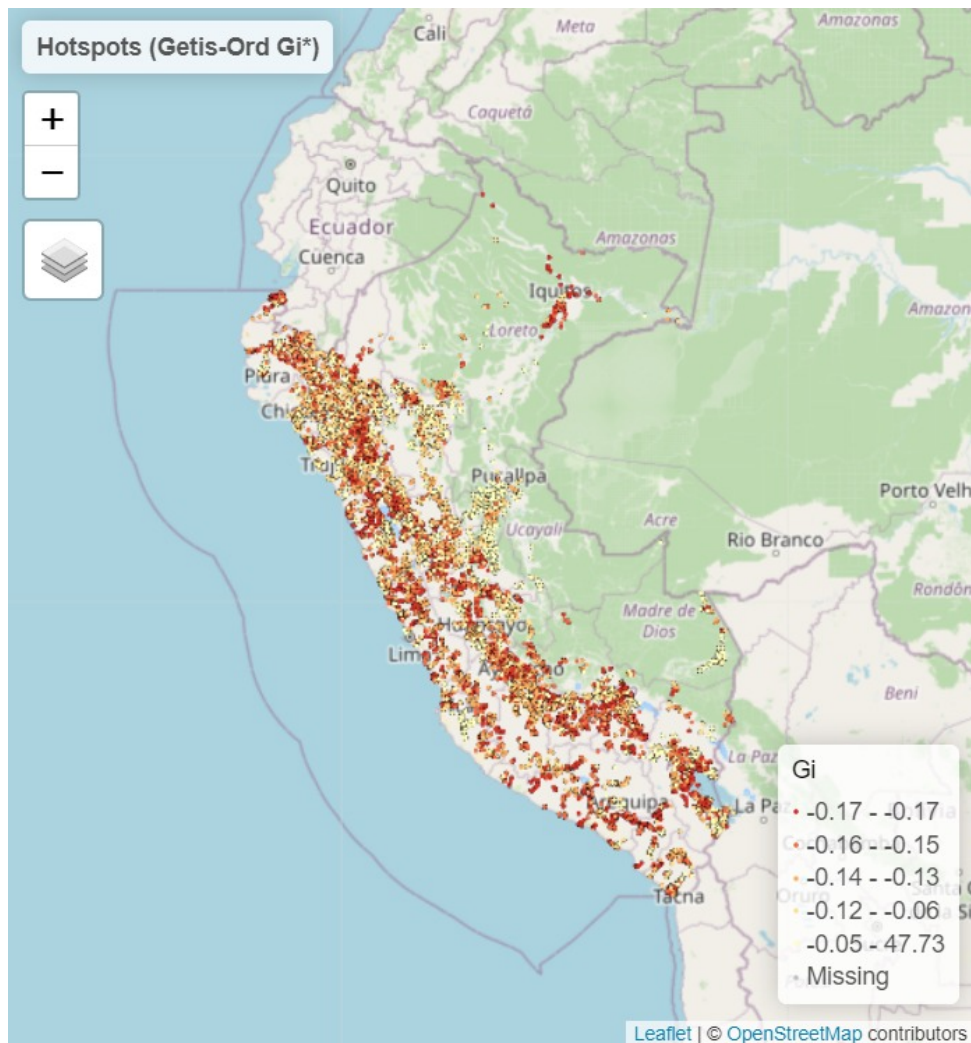


Figura 4: Mapa de Hotspots

identifica áreas de concentración significativas (hotspots). Las **matrices de pesos espaciales** son el punto de partida para todos estos cálculos, ya que definen la estructura de interacción entre las unidades del territorio.

Anexos

Código en R para el análisis espacial

```
# =====  
# APP SHINY - ANÁLISIS ESPACIAL DEFINITIVO (v3 - Estilo)  
# =====  
  
library(shiny)  
library(dplyr)  
library(sf)  
library(spdep)  
library(tmap)  
library(rnaturalearth)  
library(rnaturalearthdata)  
library(readxl)  
library(readr)  
  
# Opciones para visualización  
options(scipen = 999)  
  
# --- Modo mapa  
tmap_mode("view")  
  
# =====  
# CARGA Y LIMPIEZA DE DATOS INICIAL  
# =====  
  
# --- OPCIÓN A (Recomendada si el archivo local es REALMENTE un .xlsx)  
datos <- read_excel("C:/Users/LENOVO/Desktop/UNIVERSIDAD/X SEMESTRE/ESTADÍSTICA ESPACIAL/datos.xlsx")  
  
# Limpieza inicial: eliminar filas con coordenadas vacías  
datos <- datos %>%  
  filter(!is.na(LATITUD), !is.na(LONGITUD))  
  
# Convertir a objeto espacial (Base de datos completa)  
sf_data <- st_as_sf(datos,
```

```

        coords = c("LONGITUD", "LATITUD"),
        crs = 4326)

# Mapa base del Perú
peru <- ne_countries(scale = "medium", country = "Peru", returnclass = "sf")

# =====
# INTERFAZ (Sin Cambios)
# =====
ui <- fluidPage(
  titlePanel(" Análisis Espacial - Base CARATULA.xlsx"),

  sidebarLayout(
    sidebarPanel(
      selectInput("tema", "Seleccione un análisis:",
        choices = c("1. Matriz de Pesos" = "pesos",
                     "2. Índice de Moran" = "moran",
                     "3. Índice de Geary" = "geary",
                     "4. Hotspots (Getis-Ord Gi*)" = "hotspot")),
      selectInput("variable", "Variable a analizar:",
        choices = names(datos),
        selected = "P104_SUP_ha"),
      helpText(" Elija una variable numérica sin valores vacíos.")
    ),

    mainPanel(
      tmapOutput("mapa", height = "600px"),
      verbatimTextOutput("resultado")
    )
  )
)

# =====
# SERVIDOR
# =====
server <- function(input, output, session) {

```

```

# Reactive para preparar y limpiar los datos según la variable seleccionada
sf_cleaned_data <- reactive({
  req(input$variable)

  # 1. Extraer y limpiar la variable
  v_raw <- datos[[input$variable]]
  v <- suppressWarnings(as.numeric(v_raw))

  # Índices de las filas que tienen COORDENADAS VÁLIDAS Y VARIABLE VÁLIDA
  idx_valid <- which(!is.na(v))

  # 2. Filtrar el objeto sf (creado globalmente) y el vector de valores
  sf_use <- sf_data[idx_valid, ]
  v_clean <- v[idx_valid]

  # Validación mínima para análisis espacial
  if (length(v_clean) < 3) {
    return(NULL)
  }

  # Añadir la variable limpia para usarla en tm_dots
  sf_use$var_clean <- v_clean

  list(sf_use = sf_use, v = v_clean)
})

# La matriz de vecinos ahora depende del subset limpio (sf_cleaned_data)
vecinos <- reactive({
  data_list <- sf_cleaned_data()
  req(data_list)

  coords <- st_coordinates(data_list$sf_use)
  nb <- knn2nb(knearneigh(coords, k = 5))
  nb
})

# La matriz de pesos depende de los vecinos correctos

```

```

pesos <- reactive({
  nb2listw(vecinos(), style = "W")
})

output$mapa <- renderTmap({
  data_list <- sf_cleaned_data()

  # Manejo de error si los datos no son válidos
  if (is.null(data_list)) {
    output$resultado <- renderPrint(" Error: No hay suficientes datos válidos (al m
    return(tm_shape(peru) + tm_polygons(col = "gray90") + tm_text("NAME", size = 0.
  }

  # Extraer los datos limpios y los pesos
  sf_use <- data_list$sf_use
  v <- data_list$v
  w <- pesos()
  var_plot <- "var_clean"

  # =====
  # TEMA 1: MATRIZ DE PESOS
  # =====
  if (input$tema == "pesos") {
    output$resultado <- renderPrint({
      cat("Matriz de pesos creada con vecinos k = 5.\n")
      print(summary(pesos()))
    })

    tm_shape(peru) +
      tm_polygons(col = "gray90") +
      tm_shape(sf_use) +
      tm_dots(col = "#00A65A", size = 0.2) + # COLOR Y TAMAÑO AUMENTADOS
      tm_layout(title = "Matriz de Pesos (Vecindad k=5)")
  }

  # =====
  # TEMA 2: ÍNDICE DE MORAN

```

```

# =====
else if (input$tema == "moran") {
  i <- moran.test(v, w)
  output$resultado <- renderPrint(i)
  tm_shape(peru) +
    tm_polygons(col = "gray85") +
    tm_shape(sf_use) +
    tm_dots(col = var_plot, palette = "Reds", size = 0.2, style = "quantile") + #
    tm_layout(title = paste("Índice de Moran:", round(i$estimate[1], 4)))
}

# =====
# TEMA 3: ÍNDICE DE GEARY
# =====
else if (input$tema == "geary") {
  g <- geary.test(v, w)
  output$resultado <- renderPrint(g)
  tm_shape(peru) +
    tm_polygons(col = "gray85") +
    tm_shape(sf_use) +
    tm_dots(col = var_plot, palette = "Blues", size = 0.2, style = "quantile") + #
    tm_layout(title = paste("Índice de Geary:", round(g$estimate[1], 4)))
}

# =====
# TEMA 4: HOTSPOTS
# =====
else if (input$tema == "hotspot") {
  gi <- localG(v, w)
  sf_use$Gi <- as.numeric(gi)
  output$resultado <- renderPrint(summary(gi))
  tm_shape(peru) +
    tm_polygons(col = "gray85") +
    tm_shape(sf_use) +
    tm_dots(col = "Gi", palette = "RdYlBu", size = 0.2, style = "quantile") + #
    tm_layout(title = "Hotspots (Getis-Ord Gi*)")
}

```

```
    })  
  }  
  
# =====  
# EJECUTAR APP  
# =====  
shinyApp(ui, server)
```