tarea4_Columbus___criminalidad.R

DairXP

2025-10-15

```r
library(sf)
```

```
## Linking to GEOS 3.13.0, GDAL 3.10.1, PROJ 9.5.1; sf_use_s2() is TRUE
```

```r
library(spdep)
```

```
## Cargando paquete requerido: spData
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```r
library(ggplot2)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(spData)

# PASO 1: CARGA DE DATOS

ruta <- system.file("shapes/columbus.shp", package="spData")
print(paste("Ruta encontrada:", ruta))
```

```
## [1] "Ruta encontrada: "
```

```r
if(ruta == "") {
  data("columbus", package = "spData")
  if(!inherits(columbus, "sf")) {
    ruta <- system.file("shapes/columbus.gpkg", package="spData")
    if(ruta != "") {
      col_sf <- st_read(ruta)
    } else {
      data("columbus", package = "spdep")
      col_sf <- st_as_sf(columbus)
    }
  } else {
    col_sf <- columbus
  }
} else {
  col_sf <- st_read(ruta)
}
```

```
## Reading layer `columbus' from data source
##   `C:\Users\ALDAIR\AppData\Local\R\win-library\4.4\spData\shapes\columbus.gpkg'
##   using driver `GPKG'
## Simple feature collection with 49 features and 20 fields
## Geometry type: POLYGON
## Dimension:     XY
## Bounding box:  xmin: 5.874907 ymin: 10.78863 xmax: 11.28742 ymax: 14.74245
## Projected CRS: Undefined Cartesian SRS with unknown unit
```

```r
col_sf <- st_set_crs(col_sf, NA)

print(paste("Total de vecindarios:", nrow(col_sf)))
```

```
## [1] "Total de vecindarios: 49"
```

```r
print("\nVariables disponibles:")
```

```
## [1] "\nVariables disponibles:"
```

```r
print(names(col_sf))
```

```
##  [1] "AREA"       "PERIMETER"  "COLUMBUS_"  "COLUMBUS_I" "POLYID"
##  [6] "NEIG"       "HOVAL"      "INC"        "CRIME"      "OPEN"
## [11] "PLUMB"      "DISCBD"     "X"          "Y"          "NSA"
## [16] "NSB"        "EW"         "CP"         "THOUS"      "NEIGNO"
## [21] "geom"
```

```r
# PASO 2: ESTADISTICAS DESCRIPTIVAS

print("\n\nEstadisticas de CRIMINALIDAD:")
```

```
## [1] "\n\nEstadisticas de CRIMINALIDAD:"
```

```r
print(paste("Promedio:", round(mean(col_sf$CRIME), 2), "delitos por 1000 hogares"))
```

```
## [1] "Promedio: 35.13 delitos por 1000 hogares"
```

```r
print(paste("Minimo:", round(min(col_sf$CRIME), 2)))
```

```
## [1] "Minimo: 0.18"
```

```r
print(paste("Maximo:", round(max(col_sf$CRIME), 2)))
```

```
## [1] "Maximo: 68.89"
```

```r
print(paste("Desviacion estandar:", round(sd(col_sf$CRIME), 2)))
```

```
## [1] "Desviacion estandar: 16.73"
```

```r
# PASO 3: MATRIZ DE PESOS ESPACIALES

vecinos <- poly2nb(col_sf, queen = TRUE)
print("\nResumen de vecindad:")
```

```
## [1] "\nResumen de vecindad:"
```

```r
print(summary(vecinos))
```

```
## Neighbour list object:
## Number of regions: 49
## Number of nonzero links: 236
## Percentage nonzero weights: 9.829238
## Average number of links: 4.816327
## Link number distribution:
##
##  2  3  4  5  6  7  8  9 10
##  5  9 12  5  9  3  4  1  1
## 5 least connected regions:
## 1 6 42 46 47 with 2 links
## 1 most connected region:
## 20 with 10 links
```

```r
pesos_w <- nb2listw(vecinos, style = "W", zero.policy = TRUE)

print("\nInterpretacion:")
```

```
## [1] "\nInterpretacion:"
```

```r
print("- Usamos contigüidad QUEEN (vecinos si comparten frontera o vertice)")
```

```
## [1] "- Usamos contigüidad QUEEN (vecinos si comparten frontera o vertice)"
```

```r
print("- Style W = Pesos estandarizados por fila (suman 1)")
```

```
## [1] "- Style W = Pesos estandarizados por fila (suman 1)"
```

```r
print(paste("- Promedio de vecinos por vecindario:", round(mean(card(vecinos)), 2)))
```

```
## [1] "- Promedio de vecinos por vecindario: 4.82"
```

```r
# PASO 4: INDICE DE MORAN

moran_test <- moran.test(col_sf$CRIME, pesos_w, zero.policy = TRUE)
print(moran_test)
```

```
##
##  Moran I test under randomisation
##
## data:  col_sf$CRIME
## weights: pesos_w
##
## Moran I statistic standard deviate = 5.5894, p-value = 1.139e-08
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.500188557      -0.020833333       0.008689289
```

```r
print("\nINTERPRETACION:")
```

```
## [1] "\nINTERPRETACION:"
```

```r
print(paste("Moran's I =", round(moran_test$estimate[1], 4)))
```

```
## [1] "Moran's I = 0.5002"
```

```r
print(paste("Esperado =", round(moran_test$estimate[2], 4)))
```

```
## [1] "Esperado = -0.0208"
```

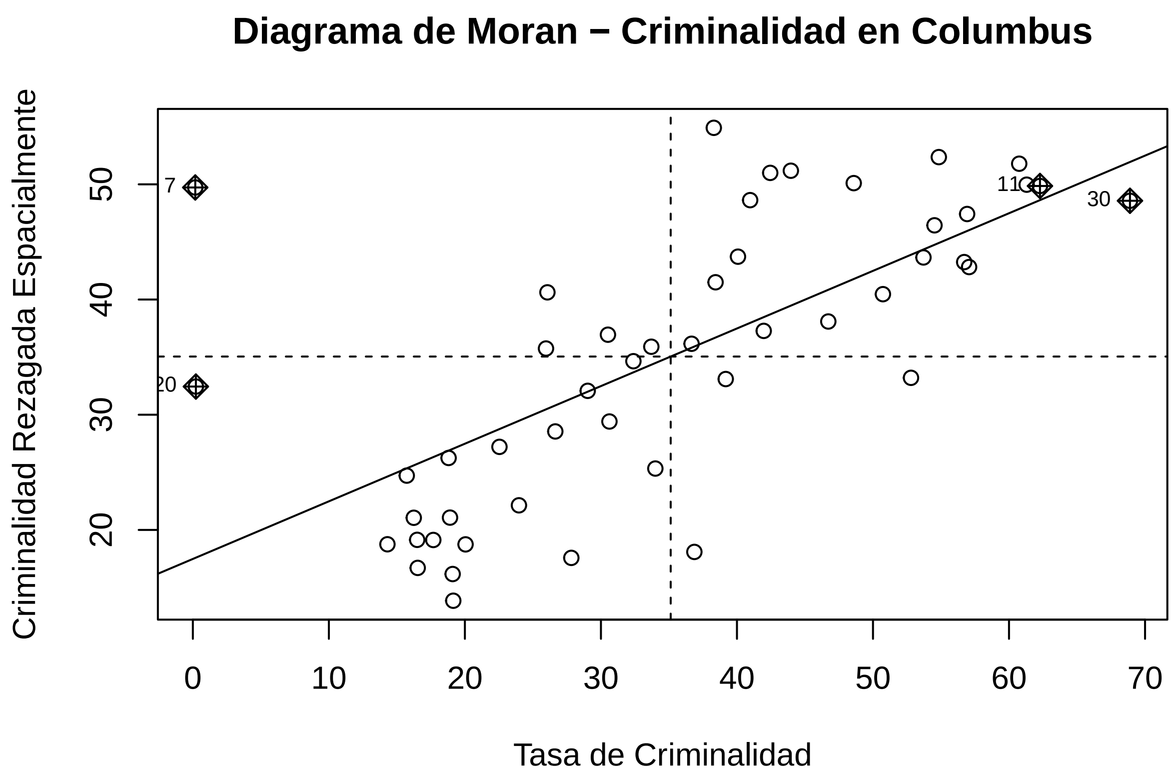
```r
print(paste("p-value =", format.pval(moran_test$p.value)))
```

```
## [1] "p-value = 1.1394e-08"
```

```r
if(moran_test$estimate[1] > 0 & moran_test$p.value < 0.05) {
  print("\nRESULTADO: AUTOCORRELACION ESPACIAL POSITIVA SIGNIFICATIVA")
  print("Los vecindarios con tasas similares de criminalidad estan juntos")
} else {
  print("\nRESULTADO: No hay patron espacial significativo")
}
```

```
## [1] "\nRESULTADO: AUTOCORRELACION ESPACIAL POSITIVA SIGNIFICATIVA"
## [1] "Los vecindarios con tasas similares de criminalidad estan juntos"
```

```r
moran.plot(col_sf$CRIME, pesos_w,
           labels = as.character(1:nrow(col_sf)),
           xlab = "Tasa de Criminalidad",
           ylab = "Criminalidad Rezagada Espacialmente",
           main = "Diagrama de Moran - Criminalidad en Columbus")
```




```r
# PASO 5: INDICE DE GEARY
geary_test <- geary.test(col_sf$CRIME, pesos_w, zero.policy = TRUE)
print(geary_test)
```

```
##
##  Geary C test under randomisation
##
## data:  col_sf$CRIME
## weights: pesos_w
##
## Geary C statistic standard deviate = 4.7431, p-value = 1.053e-06
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##       0.540528203       1.000000000       0.009384264
```

```r
print("\nINTERPRETACION:")
```

```
## [1] "\nINTERPRETACION:"
```

```r
print(paste("Geary's C =", round(geary_test$estimate[1], 4)))
```

```
## [1] "Geary’s C = 0.5405"
```

```r
print(paste("Esperado = 1.0000"))
```

```
## [1] "Esperado = 1.0000"
```

```r
print(paste("p-value =", format.pval(geary_test$p.value)))
```

```
## [1] "p-value = 1.0526e-06"
```

```r
if(geary_test$estimate[1] < 1 & geary_test$p.value < 0.05) {
  print("\nRESULTADO: AUTOCORRELACION ESPACIAL POSITIVA")
  print("Geary C < 1: Los vecinos tienen valores SIMILARES")
  print("Confirma los resultados de Moran")
} else {
  print("\nRESULTADO: No hay autocorrelacion")
}
```

```
## [1] "\nRESULTADO: AUTOCORRELACION ESPACIAL POSITIVA"
## [1] "Geary C < 1: Los vecinos tienen valores SIMILARES"
## [1] "Confirma los resultados de Moran"
```

```r
# PASO 6: HOTSPOTS - LISA

lisa <- localmoran(col_sf$CRIME, pesos_w, zero.policy = TRUE)
col_sf$lisa_I <- lisa[,1]
col_sf$lisa_pval <- lisa[,5]

quadrant <- vector(mode = "numeric", length = nrow(lisa))
m.crime <- col_sf$CRIME - mean(col_sf$CRIME)
m.local <- lag.listw(pesos_w, col_sf$CRIME, zero.policy = TRUE)
m.local <- m.local - mean(m.local, na.rm = TRUE)

signif <- 0.05
quadrant[m.crime > 0 & m.local > 0] <- 1
quadrant[m.crime < 0 & m.local < 0] <- 2
quadrant[m.crime < 0 & m.local > 0] <- 3
quadrant[m.crime > 0 & m.local < 0] <- 4
quadrant[col_sf$lisa_pval > signif] <- 0

col_sf$quadrant <- factor(quadrant,
                          levels = 0:4,
                          labels = c("No significativo",
                                     "Alto-Alto (Hotspot)",
                                     "Bajo-Bajo (Coldspot)",
                                     "Bajo-Alto (Outlier)",
                                     "Alto-Bajo (Outlier)"))

print("\nDEFINICIONES:")
```

```
## [1] "\nDEFINICIONES:"
```

```r
print("1. Alto-Alto (HOTSPOT): Alta criminalidad rodeado de alta criminalidad")
```

```
## [1] "1. Alto-Alto (HOTSPOT): Alta criminalidad rodeado de alta criminalidad"
```

```r
print("2. Bajo-Bajo (COLDSPOT): Baja criminalidad rodeado de baja criminalidad")
```

```
## [1] "2. Bajo-Bajo (COLDSPOT): Baja criminalidad rodeado de baja criminalidad"
```

```r
print("3. Bajo-Alto: Baja criminalidad rodeado de alta criminalidad")
```

```
## [1] "3. Bajo-Alto: Baja criminalidad rodeado de alta criminalidad"
```

```r
print("4. Alto-Bajo: Alta criminalidad rodeado de baja criminalidad")
```

```
## [1] "4. Alto-Bajo: Alta criminalidad rodeado de baja criminalidad"
```

```r
tabla_clusters <- table(col_sf$quadrant)
print("\nDistribucion de clusters:")
```

```
## [1] "\nDistribucion de clusters:"
```

```r
print(tabla_clusters)
```

```
## 
##     No significativo  Alto-Alto (Hotspot) Bajo-Bajo (Coldspot) 
##                   34                   11                    4 
##   Bajo-Alto (Outlier)  Alto-Bajo (Outlier) 
##                    0                    0 
```

```r
tabla_hotspots <- col_sf %>%
  st_drop_geometry() %>%
  filter(quadrant != "No significativo") %>%
  select(CRIME, quadrant, lisa_I, lisa_pval) %>%
  arrange(desc(abs(lisa_I)))

print("\nVecindarios con clusters significativos:")
```

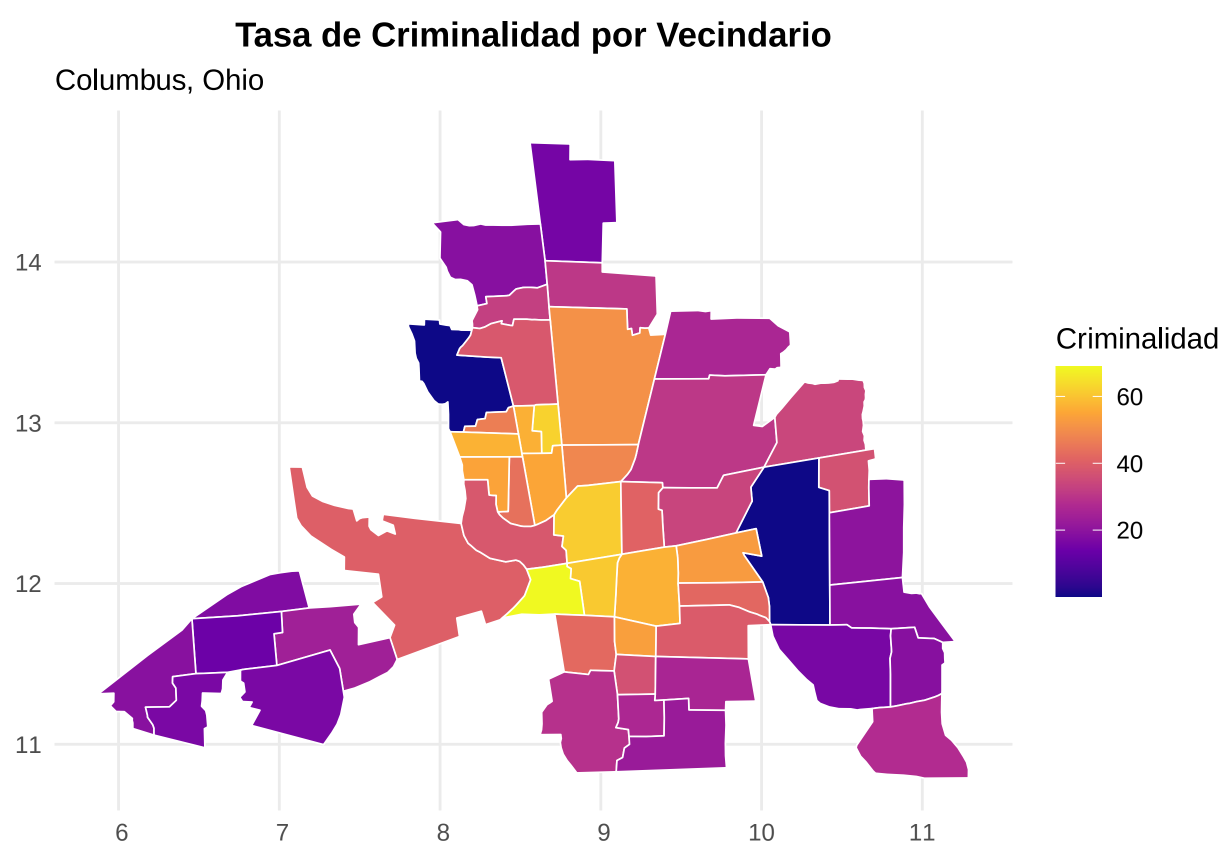
```
## [1] "\nVecindarios con clusters significativos:"
```

```r
print(tabla_hotspots)
```

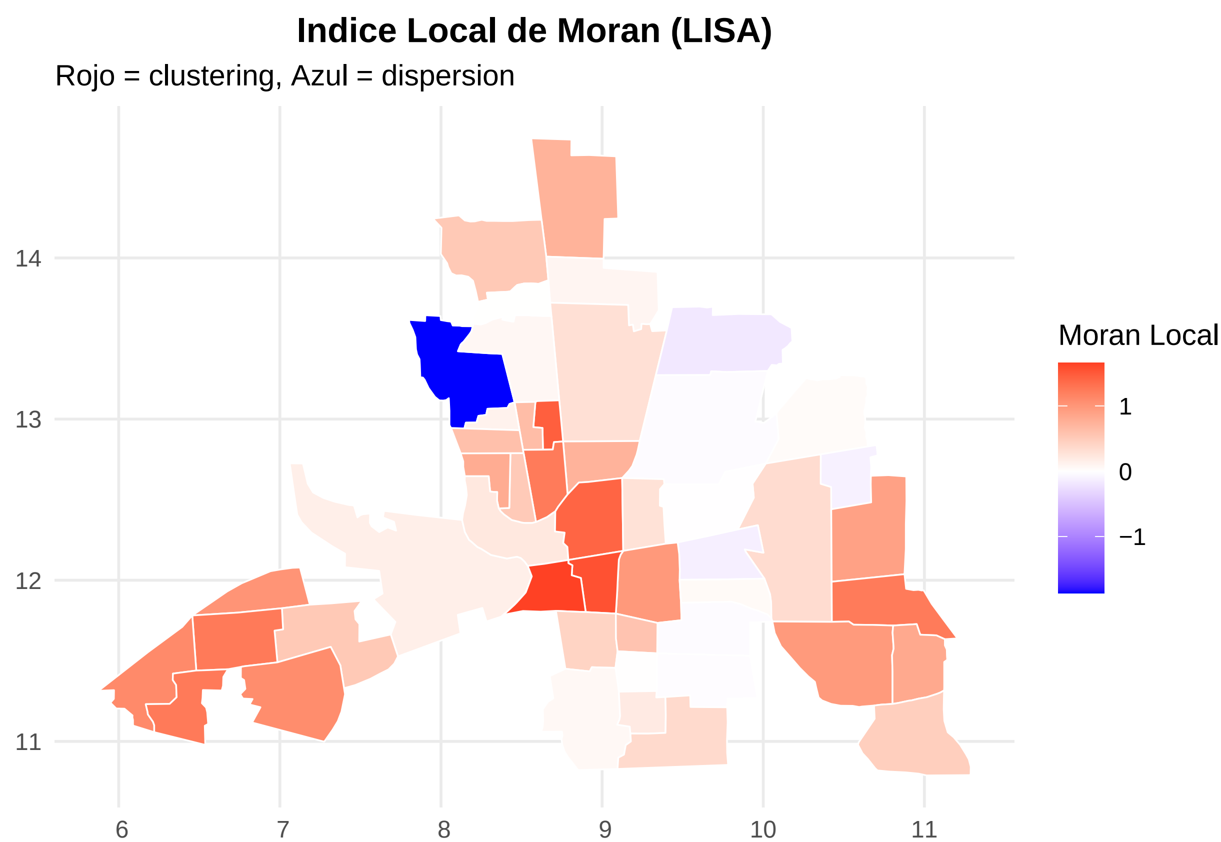
```
##       CRIME            quadrant    lisa_I    lisa_pval
## 1  68.89204  Alto-Alto (Hotspot) 1.6551715 0.0386132784
## 2  60.75045  Alto-Alto (Hotspot) 1.5566245 0.0028206853
## 3  62.27545  Alto-Alto (Hotspot) 1.4579154 0.0278562842
## 4  61.29917  Alto-Alto (Hotspot) 1.4163442 0.0037803826
```

```
## 5  14.30556 Bajo-Bajo (Coldspot) 1.2432425 0.0169409917
## 6  19.14559 Bajo-Bajo (Coldspot) 1.2398550 0.0070262965
## 7  54.83871  Alto-Alto (Hotspot) 1.2387718 0.0010291539
## 8  19.10086 Bajo-Bajo (Coldspot) 1.1079465 0.0392923036
## 9  56.91978  Alto-Alto (Hotspot) 0.9769529 0.0105894533
## 10 16.24130 Bajo-Bajo (Coldspot) 0.9692979 0.0404268156
## 11 48.58549  Alto-Alto (Hotspot) 0.7348439 0.0173542681
## 12 43.96249  Alto-Alto (Hotspot) 0.5170513 0.0442421009
## 13 42.44508  Alto-Alto (Hotspot) 0.4232470 0.0129319698
## 14 40.96974  Alto-Alto (Hotspot) 0.2875311 0.0346590965
## 15 38.29787  Alto-Alto (Hotspot) 0.2285270 0.0007775888
```

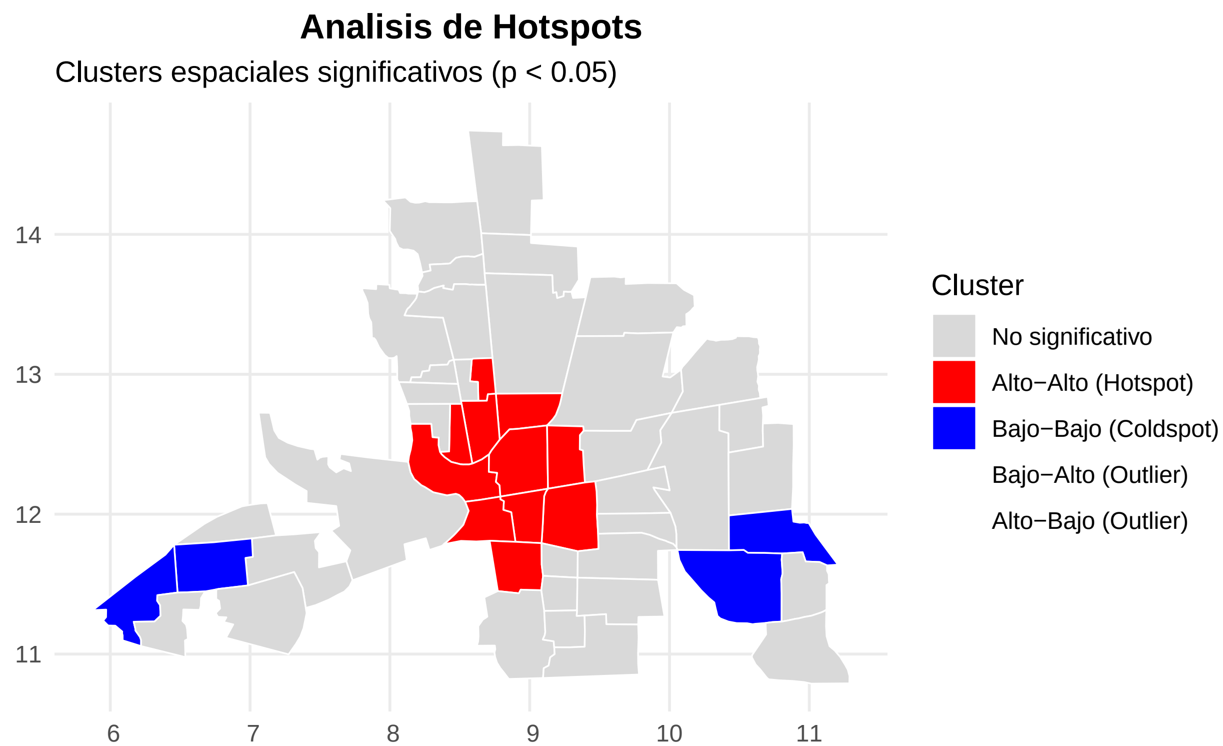
```r
# PASO 7: VISUALIZACIONES

mapa_crime <- ggplot(col_sf) +
  geom_sf(aes(fill = CRIME), color = "white", size = 0.3) +
  scale_fill_viridis_c(option = "plasma", name = "Criminalidad") +
  labs(title = "Tasa de Criminalidad por Vecindario",
       subtitle = "Columbus, Ohio") +
  theme_minimal() +
  theme(legend.position = "right",
        plot.title = element_text(hjust = 0.5, face = "bold"))

print(mapa_crime)
```



```r
mapa_lisa <- ggplot(col_sf) +
  geom_sf(aes(fill = lisa_I), color = "white", size = 0.3) +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
                       midpoint = 0, name = "Moran Local") +
  labs(title = "Indice Local de Moran (LISA)",
       subtitle = "Rojo = clustering, Azul = dispersion") +
  theme_minimal() +
  theme(legend.position = "right",
        plot.title = element_text(hjust = 0.5, face = "bold"))

print(mapa_lisa)
```




```r
mapa_hotspots <- ggplot(col_sf) +
  geom_sf(aes(fill = quadrant), color = "white", size = 0.3) +
  scale_fill_manual(values = c("grey85", "red", "blue", "lightpink", "lightskyblue"),
                    name = "Cluster",
                    drop = FALSE) +
  labs(title = "Analisis de Hotspots",
       subtitle = "Clusters espaciales significativos (p < 0.05)") +
  theme_minimal() +
  theme(legend.position = "right",
        plot.title = element_text(hjust = 0.5, face = "bold"))

print(mapa_hotspots)
```




```r
# PASO 8: RESUMEN FINAL

resultados <- data.frame(
  Indicador = c("Moran's I", "Geary's C"),
  Valor = c(round(moran_test$estimate[1], 4), round(geary_test$estimate[1], 4)),
  Esperado = c(round(moran_test$estimate[2], 4), 1.0000),
  p_value = c(format.pval(moran_test$p.value), format.pval(geary_test$p.value)),
  Significativo = c(
    ifelse(moran_test$p.value < 0.05, "SI", "NO"),
    ifelse(geary_test$p.value < 0.05, "SI", "NO")
  )
)

print(resultados)
```

```
##                      Indicador  Valor Esperado    p_value Significativo
## Moran I statistic Moran's I 0.5002  -0.0208 1.1394e-08            SI
## Geary C statistic Geary's C 0.5405   1.0000 1.0526e-06            SI
```

```r
cluster_summary <- col_sf %>%
  st_drop_geometry() %>%
  group_by(quadrant) %>%
  summarise(
    n = n(),
    crime_promedio = round(mean(CRIME), 2)
```

```r
  )

print("\nResumen por tipo de cluster:")
```

```
## [1] "\nResumen por tipo de cluster:"
```

```r
print(cluster_summary)
```

```
## # A tibble: 3 x 3
##   quadrant                  n crime_promedio
##   <fct>                 <int>          <dbl>
## 1 No significativo         34           31.6
## 2 Alto-Alto (Hotspot)      11           52.7
## 3 Bajo-Bajo (Coldspot)      4           17.2
```