Matrices_Moran_Geary_Hotspots.R

ASUS

2025-10-15

```
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')`

## Cargando paquete requerido: sf

## Linking to GEOS 3.13.1, GDAL 3.11.0, PROJ 9.6.0; sf_use_s2() is TRUE
```

```
library(sf)
library(ggplot2)
library(viridis)
```

```
## Cargando paquete requerido: viridisLite
```

```
# ============================================================================
# 1. DATOS ESPACIALES DE EJEMPLO
# ============================================================================

set.seed(123)

# Crear una cuadrícula de 10x10 polígonos
n_rows <- 10
n_cols <- 10
cell_size <- 1

# Generar coordenadas de los centroides
coords <- expand.grid(x = 1:n_cols, y = 1:n_rows)

# Crear polígonos (cuadrados)
polys <- list()
for(i in 1:nrow(coords)) {
  x <- coords$x[i]
  y <- coords$y[i]

  poly_coords <- matrix(c(
```

```r
    x - 0.5, y - 0.5,
    x + 0.5, y - 0.5,
    x + 0.5, y + 0.5,
    x - 0.5, y + 0.5,
    x - 0.5, y - 0.5
  ), ncol = 2, byrow = TRUE)

  polys[[i]] <- st_polygon(list(poly_coords))
}

# Crear objeto sf
spatial_data <- st_sf(
  id = 1:nrow(coords),
  geometry = st_sfc(polys),
  crs = NA
)

# Agregar variable de interés con autocorrelación espacial
# Crear clusters de valores altos y bajos
spatial_data$valor <- rnorm(nrow(spatial_data), mean = 50, sd = 10)

# Crear hotspot en la esquina superior derecha
hotspot_ids <- which(coords$x >= 7 & coords$y >= 7)
spatial_data$valor[hotspot_ids] <- spatial_data$valor[hotspot_ids] + 30

# Crear coldspot en la esquina inferior izquierda
coldspot_ids <- which(coords$x <= 4 & coords$y <= 4)
spatial_data$valor[coldspot_ids] <- spatial_data$valor[coldspot_ids] - 20

# ============================================================================
# 2. MATRICES DE PESOS ESPACIALES
# ============================================================================

cat("\n=== MATRICES DE PESOS ESPACIALES ===\n")
```

```
##
## === MATRICES DE PESOS ESPACIALES ===
```

```r
# 2.1 Matriz de vecindad por contigüidad (Queen)
nb_queen <- poly2nb(spatial_data, queen = TRUE)
cat("\nVecinos por contigüidad Queen (8 vecinos):\n")
```

```
##
## Vecinos por contigüidad Queen (8 vecinos):
```

```r
print(summary(nb_queen))
```

```
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 684
## Percentage nonzero weights: 6.84
## Average number of links: 6.84
```

```
## Link number distribution:
##
##  3  5  8
##  4 32 64
## 4 least connected regions:
## 1 10 91 100 with 3 links
## 64 most connected regions:
## 12 13 14 15 16 17 18 19 22 23 24 25 26 27 28 29 32 33 34 35 36 37 38 39 42 43 44 45 46 47 48 49 52 5
```

```r
# 2.2 Matriz de vecindad por contigüidad (Rook)
nb_rook <- poly2nb(spatial_data, queen = FALSE)
cat("\nVecinos por contigüidad Rook (4 vecinos):\n")
```

```
##
## Vecinos por contigüidad Rook (4 vecinos):
```

```r
print(summary(nb_rook))
```

```
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 360
## Percentage nonzero weights: 3.6
## Average number of links: 3.6
## Link number distribution:
##
##  2  3  4
##  4 32 64
## 4 least connected regions:
## 1 10 91 100 with 2 links
## 64 most connected regions:
## 12 13 14 15 16 17 18 19 22 23 24 25 26 27 28 29 32 33 34 35 36 37 38 39 42 43 44 45 46 47 48 49 52 5
```

```r
# 2.3 Convertir a matriz de pesos (row-standardized)
W_queen <- nb2listw(nb_queen, style = "W", zero.policy = TRUE)
W_rook <- nb2listw(nb_rook, style = "W", zero.policy = TRUE)

# 2.4 Matriz de pesos por distancia (k-vecinos más cercanos)
coords_centroids <- st_coordinates(st_centroid(spatial_data))
```

```
## Warning: st_centroid assumes attributes are constant over geometries
```

```r
k <- 4
nb_knn <- knn2nb(knearneigh(coords_centroids, k = k))
W_knn <- nb2listw(nb_knn, style = "W")
cat("\nVecinos por k-NN (k=4):\n")
```

```
##
## Vecinos por k-NN (k=4):
```

```r
print(summary(nb_knn))
```

```
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 400
## Percentage nonzero weights: 4
## Average number of links: 4
## Non-symmetric neighbours list
## Link number distribution:
##
##   4
## 100
## 100 least connected regions:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
## 100 most connected regions:
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
```

```r
# ============================================================================
# 3. ÍNDICE DE MORAN (AUTOCORRELACIÓN ESPACIAL GLOBAL)
# ============================================================================

cat("\n\n=== ÍNDICE DE MORAN ===\n")
```

```
##
##
## === ÍNDICE DE MORAN ===
```

```r
# Calcular el Índice de Moran I
moran_test <- moran.test(spatial_data$valor, W_queen)
cat("\nÍndice de Moran I:\n")
```

```
##
## Índice de Moran I:
```

```r
print(moran_test)
```

```
##
##  Moran I test under randomisation
##
## data:  spatial_data$valor
## weights: W_queen
##
## Moran I statistic standard deviate = 11.306, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.593336420      -0.010101010       0.002848939
```

```r
# Interpretación
cat("\nInterpretación del Índice de Moran:\n")
```

```
##
## Interpretación del Índice de Moran:
```

```r
cat(sprintf("Moran's I = %.4f\n", moran_test$estimate[1]))
```

```
## Moran's I = 0.5933
```

```r
cat(sprintf("p-value = %.4f\n", moran_test$p.value))
```

```
## p-value = 0.0000
```

```r
if(moran_test$p.value < 0.05) {
  if(moran_test$estimate[1] > 0) {
    cat("Conclusión: Existe autocorrelación espacial POSITIVA significativa\n")
    cat("(valores similares tienden a agruparse espacialmente)\n")
  } else {
    cat("Conclusión: Existe autocorrelación espacial NEGATIVA significativa\n")
    cat("(valores diferentes tienden a estar juntos)\n")
  }
} else {
  cat("Conclusión: No hay evidencia de autocorrelación espacial\n")
}
```

```
## Conclusión: Existe autocorrelación espacial POSITIVA significativa
## (valores similares tienden a agruparse espacialmente)
```

```r
# Test de permutación de Monte Carlo
moran_mc <- moran.mc(spatial_data$valor, W_queen, nsim = 999)
cat("\n\nTest de Monte Carlo (999 permutaciones):\n")
```

```
##
##
## Test de Monte Carlo (999 permutaciones):
```

```r
print(moran_mc)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  spatial_data$valor
## weights: W_queen
## number of simulations + 1: 1000
##
## statistic = 0.59334, observed rank = 1000, p-value = 0.001
## alternative hypothesis: greater
```

```r
# ============================================================================
# 4. ÍNDICE DE GEARY (AUTOCORRELACIÓN ESPACIAL GLOBAL)
# ============================================================================

cat("\n\n=== ÍNDICE DE GEARY ===\n")
```

```
##
##
## === ÍNDICE DE GEARY ===
```

```r
# Calcular el Índice de Geary C
geary_test <- geary.test(spatial_data$valor, W_queen)
cat("\nÍndice de Geary C:\n")
```

```
##
## Índice de Geary C:
```

```r
print(geary_test)
```

```
##
##  Geary C test under randomisation
##
## data:  spatial_data$valor
## weights: W_queen
##
## Geary C statistic standard deviate = 10.954, p-value < 2.2e-16
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##       0.395053834       1.000000000       0.003050118
```

```r
# Interpretación
cat("\nInterpretación del Índice de Geary:\n")
```

```
##
## Interpretación del Índice de Geary:
```

```r
cat(sprintf("Geary's C = %.4f\n", geary_test$estimate[1]))
```

```
## Geary's C = 0.3951
```

```r
cat(sprintf("p-value = %.4f\n", geary_test$p.value))
```

```
## p-value = 0.0000
```

```r
cat("\nNota: Geary's C varía de 0 a 2+\n")
```

```
##
## Nota: Geary's C varía de 0 a 2+
```

```r
cat("C < 1: autocorrelación positiva\n")
```

```
## C < 1: autocorrelación positiva
```

```r
cat("C = 1: aleatoriedad espacial\n")
```

```
## C = 1: aleatoriedad espacial
```

```r
cat("C > 1: autocorrelación negativa\n")
```

```
## C > 1: autocorrelación negativa
```

```r
# Test de permutación de Monte Carlo
geary_mc <- geary.mc(spatial_data$valor, W_queen, nsim = 999)
cat("\n\nTest de Monte Carlo (999 permutaciones):\n")
```

```
##
##
## Test de Monte Carlo (999 permutaciones):
```

```r
print(geary_mc)
```

```
##
##  Monte-Carlo simulation of Geary C
##
## data:  spatial_data$valor
## weights: W_queen
## number of simulations + 1: 1000
##
## statistic = 0.39505, observed rank = 1, p-value = 0.001
## alternative hypothesis: greater
```

```r
# ============================================================================
# 5. ANÁLISIS DE HOTSPOTS (LISA - Local Indicators of Spatial Association)
# ============================================================================

cat("\n\n=== ANÁLISIS DE HOTSPOTS (LISA) ===\n")
```

```
##
##
## === ANÁLISIS DE HOTSPOTS (LISA) ===
```

```r
# Calcular índices de Moran locales
local_moran <- localmoran(spatial_data$valor, W_queen)

# Agregar resultados al dataset
spatial_data$local_I <- local_moran[, 1]  # Índice local
spatial_data$local_pval <- local_moran[, 5]  # p-valor

# Estandarizar valores para clasificación
spatial_data$valor_std <- scale(spatial_data$valor)
spatial_data$lag_valor_std <- lag.listw(W_queen, spatial_data$valor_std)

# Clasificar en categorías LISA
```

```r
spatial_data$lisa_cat <- "No significativo"
sig_level <- 0.05

# High-High (Hotspots)
spatial_data$lisa_cat[spatial_data$valor_std > 0 &
                        spatial_data$lag_valor_std > 0 &
                        spatial_data$local_pval < sig_level] <- "High-High (Hotspot)"

# Low-Low (Coldspots)
spatial_data$lisa_cat[spatial_data$valor_std < 0 &
                        spatial_data$lag_valor_std < 0 &
                        spatial_data$local_pval < sig_level] <- "Low-Low (Coldspot)"

# High-Low (Outliers)
spatial_data$lisa_cat[spatial_data$valor_std > 0 &
                        spatial_data$lag_valor_std < 0 &
                        spatial_data$local_pval < sig_level] <- "High-Low (Outlier)"

# Low-High (Outliers)
spatial_data$lisa_cat[spatial_data$valor_std < 0 &
                        spatial_data$lag_valor_std > 0 &
                        spatial_data$local_pval < sig_level] <- "Low-High (Outlier)"

# Resumen de clusters
cat("\nDistribución de clusters LISA:\n")
```

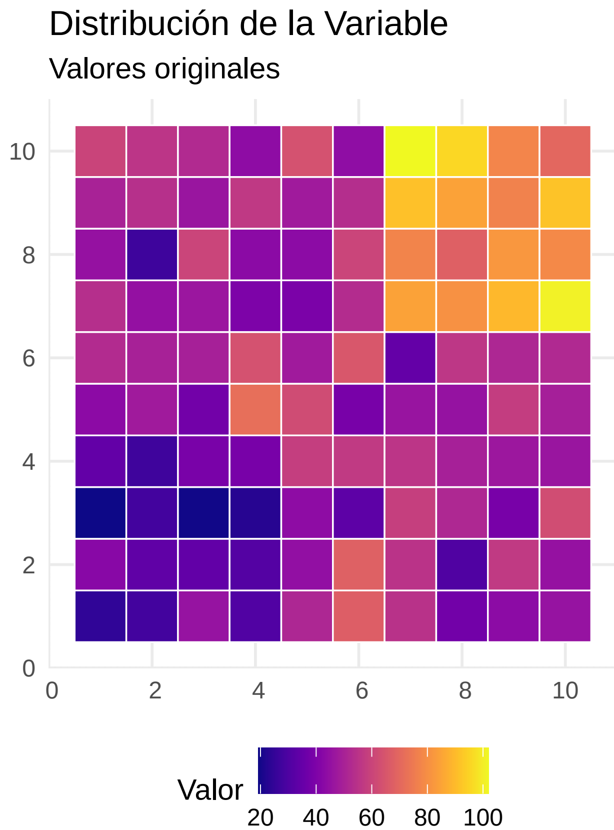
```
##
## Distribución de clusters LISA:
```

```r
print(table(spatial_data$lisa_cat))
```

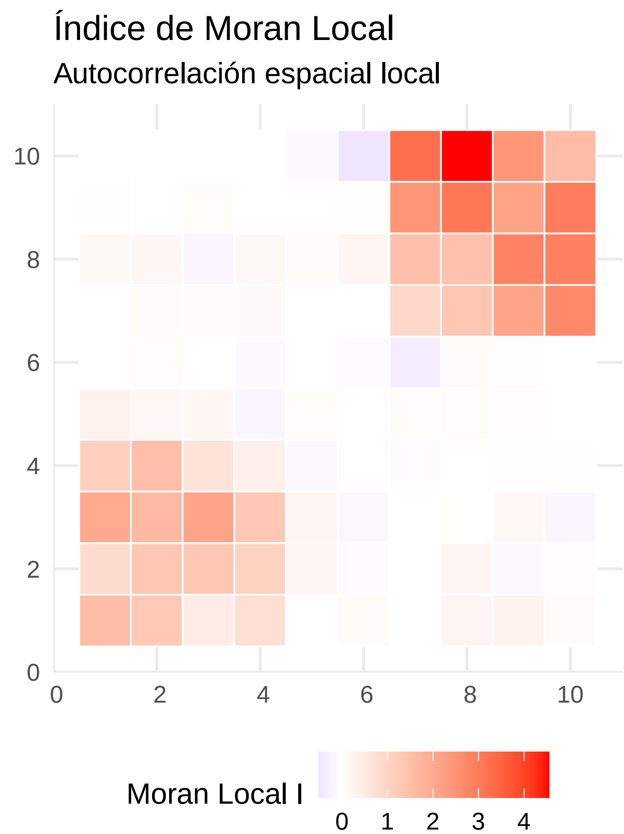
```
##
## High-High (Hotspot)  Low-High (Outlier)  Low-Low (Coldspot)    No significativo
##                  16                   3                  13                  68
```

```r
# ============================================================================
# 6. VISUALIZACIONES
# ============================================================================

# Gráfico 1: Variable original
p1 <- ggplot(spatial_data) +
  geom_sf(aes(fill = valor), color = "white", size = 0.3) +
  scale_fill_viridis(option = "plasma", name = "Valor") +
  labs(title = "Distribución de la Variable",
       subtitle = "Valores originales") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Gráfico 2: Índice de Moran Local
p2 <- ggplot(spatial_data) +
  geom_sf(aes(fill = local_I), color = "white", size = 0.3) +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red",
```

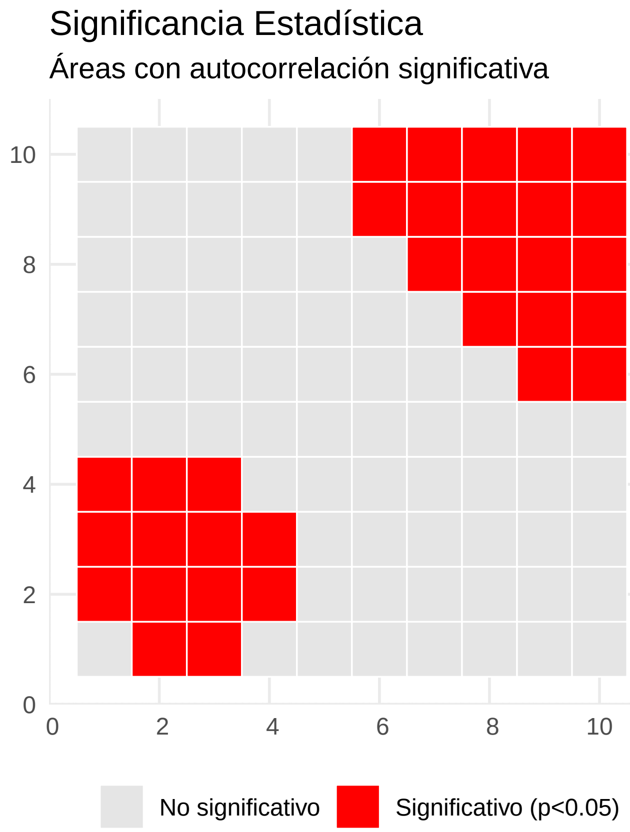
```r
                       midpoint = 0, name = "Moran Local I") +
  labs(title = "Índice de Moran Local",
       subtitle = "Autocorrelación espacial local") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Gráfico 3: Significancia estadística
spatial_data$significativo <- ifelse(spatial_data$local_pval < 0.05,
                                     "Significativo (p<0.05)",
                                     "No significativo")
p3 <- ggplot(spatial_data) +
  geom_sf(aes(fill = significativo), color = "white", size = 0.3) +
  scale_fill_manual(values = c("Significativo (p<0.05)" = "red",
                               "No significativo" = "gray90"),
                    name = "") +
  labs(title = "Significancia Estadística",
       subtitle = "Áreas con autocorrelación significativa") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Gráfico 4: Clusters LISA (Hotspots/Coldspots)
lisa_colors <- c("High-High (Hotspot)" = "#d7191c",
                 "Low-Low (Coldspot)" = "#2b83ba",
                 "High-Low (Outlier)" = "#fdae61",
                 "Low-High (Outlier)" = "#abd9e9",
                 "No significativo" = "gray90")

p4 <- ggplot(spatial_data) +
  geom_sf(aes(fill = lisa_cat), color = "white", size = 0.3) +
  scale_fill_manual(values = lisa_colors, name = "Categoría LISA") +
  labs(title = "Clusters LISA: Hotspots y Coldspots",
       subtitle = "High-High = Hotspots, Low-Low = Coldspots") +
  theme_minimal() +
  theme(legend.position = "bottom")

# Mostrar gráficos
print(p1)
```


Distribución de la Variable
Valores originales
10
8
6
4
2
0
0
2
4
6
8
10
Valor
20
40
60
80
100

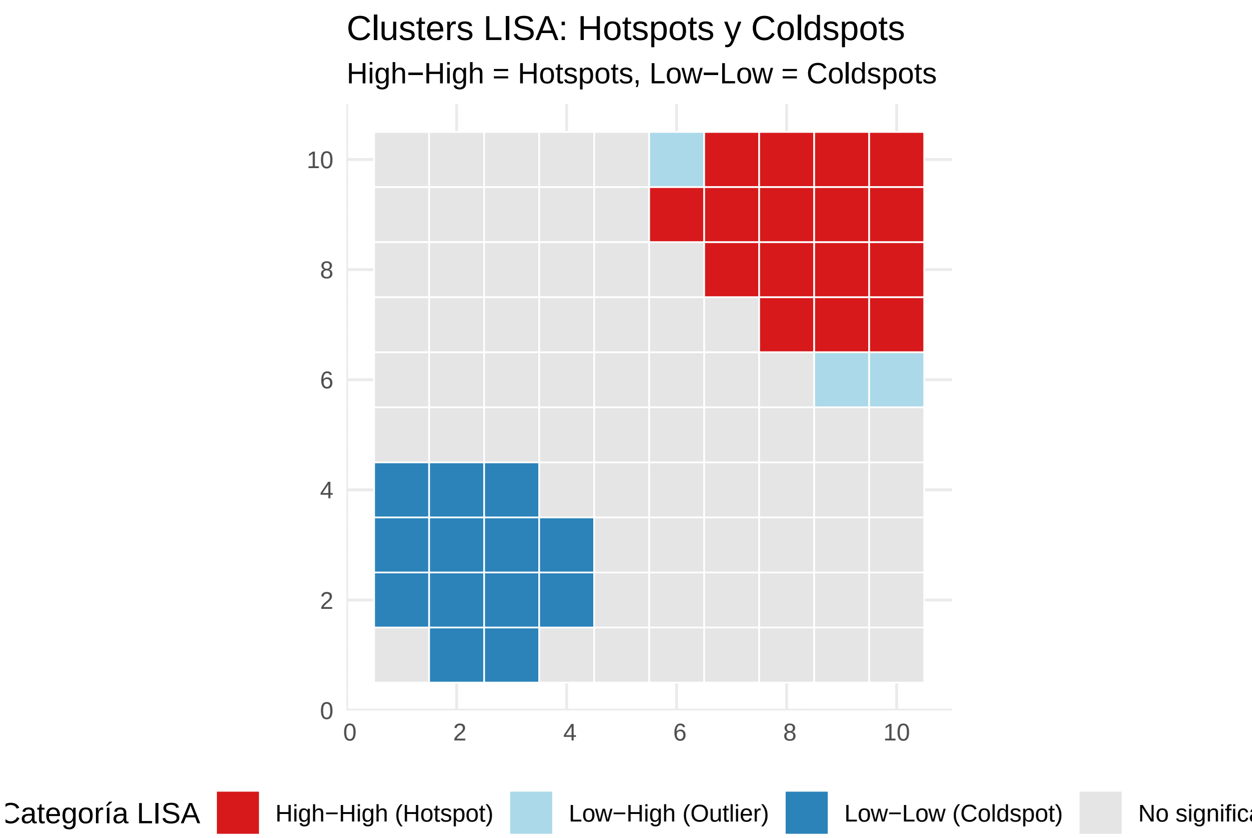
```r
print(p2)
```


Índice de Moran Local
Autocorrelación espacial local
10
8
6
4
2
0
0
2
4
6
8
10
Moran Local I
0
1
2
3
4

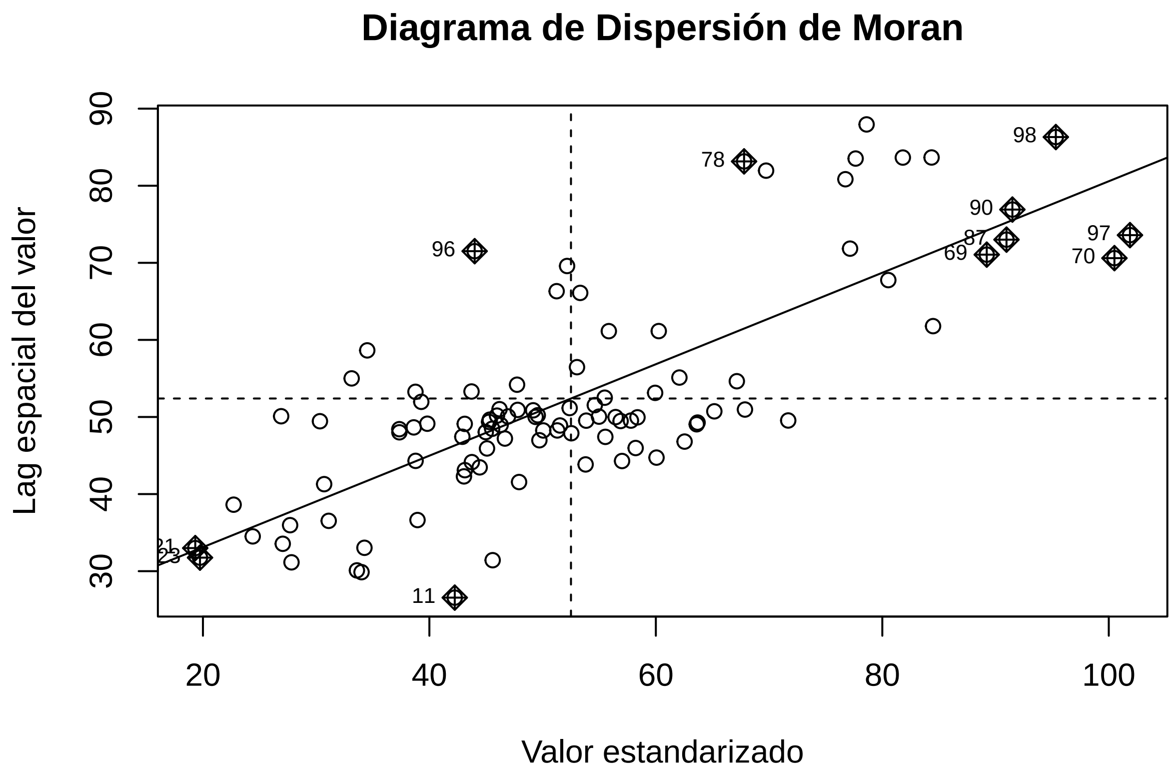
```r
print(p3)
```


Significancia Estadística
Áreas con autocorrelación significativa
10
8
6
4
2
0
0
2
4
6
8
10
No significativo
Significativo (p<0.05)

```r
print(p4)
```




```r
# Gráfico 5: Diagrama de dispersión de Moran
moran_plot <- moran.plot(spatial_data$valor, W_queen,
                         labels = as.character(spatial_data$id),
                         xlab = "Valor estandarizado",
                         ylab = "Lag espacial del valor",
                         main = "Diagrama de Dispersión de Moran")
```




```r
# ==========================================================================
# 7. EXPORTAR RESULTADOS
# ==========================================================================

# Crear resumen de resultados
resultados <- data.frame(
  id = spatial_data$id,
  valor = spatial_data$valor,
  moran_local = spatial_data$local_I,
  pvalor = spatial_data$local_pval,
  categoria_lisa = spatial_data$lisa_cat
)

cat("\n\nPrimeras filas del resultado:\n")
```

```
##
##
## Primeras filas del resultado:
```

```r
print(head(resultados, 10))
```

```
##    id    valor  moran_local      pvalor     categoria_lisa
## 1   1 24.39524  1.586413079 0.071070361   No significativo
## 2   2 27.69823  1.286903583 0.030986851 Low-Low (Coldspot)
## 3   3 45.58708  0.457411405 0.007078416 Low-Low (Coldspot)
```

```
## 4   4 30.70508  0.767024101 0.142577720   No significativo
## 5   5 51.29288  0.016145057 0.587654990   No significativo
## 6   6 67.15065  0.098055682 0.770777978   No significativo
## 7   7 54.60916 -0.006390617 0.904138788   No significativo
## 8   8 37.34939  0.213485389 0.553171433   No significativo
## 9   9 43.13147  0.276416619 0.226304060   No significativo
## 10 10 45.54338  0.088032040 0.688901499   No significativo
```

```
# Opcional: guardar resultados
# write.csv(resultados, "resultados_analisis_espacial.csv", row.names = FALSE)
# st_write(spatial_data, "datos_espaciales.shp")

cat("\n\n=== ANÁLISIS COMPLETADO ===\n")
```

```
##
##
## === ANÁLISIS COMPLETADO ===
```

```
cat("\nResumen de métricas globales:\n")
```

```
##
## Resumen de métricas globales:
```

```
cat(sprintf("- Índice de Moran I: %.4f (p = %.4f)\n",
            moran_test$estimate[1], moran_test$p.value))
```

```
## - Índice de Moran I: 0.5933 (p = 0.0000)
```

```
cat(sprintf("- Índice de Geary C: %.4f (p = %.4f)\n",
            geary_test$estimate[1], geary_test$p.value))
```

```
## - Índice de Geary C: 0.3951 (p = 0.0000)
```

```
cat(sprintf("- Hotspots detectados: %d\n",
            sum(spatial_data$lisa_cat == "High-High (Hotspot)")))
```

```
## - Hotspots detectados: 16
```

```
cat(sprintf("- Coldspots detectados: %d\n",
            sum(spatial_data$lisa_cat == "Low-Low (Coldspot)")))
```

```
## - Coldspots detectados: 13
```