```r
# ================================
#  Ejemplo completo en R
# ================================

# Instalación (solo la primera vez)
# install.packages(c("sf", "spdep", "spData", "tmap"))

library(sf)

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

library(spdep)

## Warning: package 'spdep' was built under R version 4.4.3

## Loading required package: spData

## Warning: package 'spData' was built under R version 4.4.3

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

library(spData)
library(tmap)

## Warning: package 'tmap' was built under R version 4.4.3

# [1] Matrices de Pesos Espaciales

# Usamos el dataset 'nc' (condados de Carolina del Norte)
nc <- st_read(system.file("shape/nc.shp", package="sf"), quiet = TRUE)

# Creamos la lista de vecinos basados en contigüidad (reinas)
vecinos <- poly2nb(nc)

# Creamos la matriz de pesos espaciales (estandarizados)
W <- nb2listw(vecinos, style = "W")

# Visualizamos parte de la matriz de pesos
print(W, zero.policy = TRUE)

## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 100
## Number of nonzero links: 490
## Percentage nonzero weights: 4.9
## Average number of links: 4.9
##
## Weights style: W
## Weights constants summary:
```

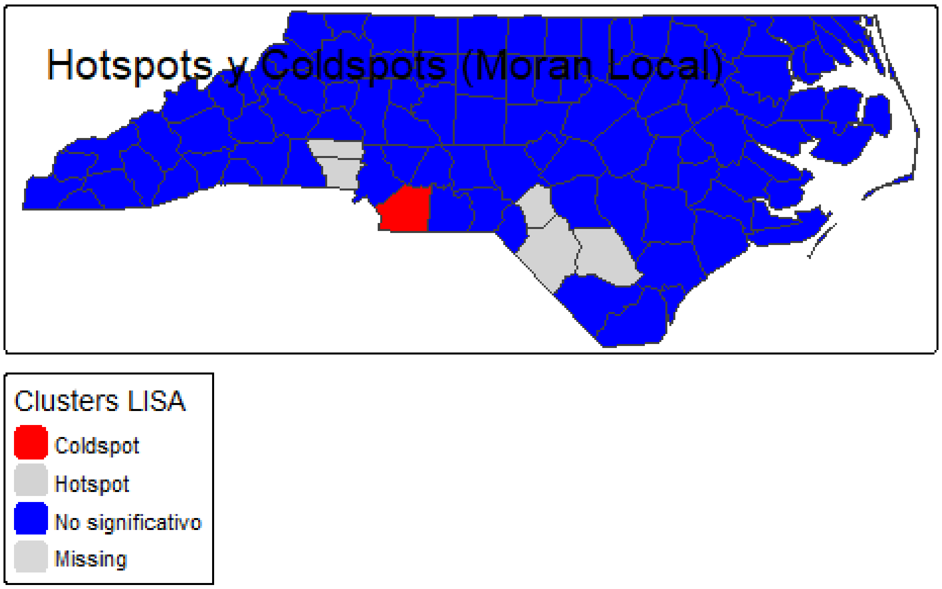
```r
##     n    nn  S0       S1       S2
## W 100 10000 100 44.65023 410.4746

# [2] Índice de Moran Global

# Variable de interés: tasa de SIDS (síndrome de muerte súbita infantil)
moran_global <- moran.test(nc$SID74, listw = W)

cat("\n=== Moran's I Global ===\n")

##
## === Moran's I Global ===

print(moran_global)

##
##  Moran I test under randomisation
##
## data:  nc$SID74
## weights: W
##
## Moran I statistic standard deviate = 2.5192, p-value = 0.00588
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       0.147740529      -0.010101010       0.003925567

# [3] Índice de Geary

geary_global <- geary.test(nc$SID74, listw = W)

cat("\n=== Geary's C ===\n")

##
## === Geary's C ===

print(geary_global)

##
##  Geary C test under randomisation
##
## data:  nc$SID74
## weights: W
##
## Geary C statistic standard deviate = 1.9591, p-value = 0.02505
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##       0.843876721       1.000000000       0.006350747

# [4] Hotspots (Moran Local / LISA)
```

```r
moran_local <- localmoran(nc$SID74, listw = W)

# Añadimos los resultados al shapefile
nc$Ii <- moran_local[,1]         # Valor de I local
nc$Z_Ii <- moran_local[,4]       # Valor Z

# Clasificamos los hotspots y coldspots
nc$cluster <- ifelse(nc$Z_Ii > 1.96, "Hotspot",
                     ifelse(nc$Z_Ii < -1.96, "Coldspot", "No significativ
o"))

# Visualizamos los clusters espaciales
tmap_mode("plot")
```

```
## ℹ tmap modes "plot" - "view"
## ℹ toggle with `tmap::ttm()`
```

```r
tm_shape(nc) +
  tm_fill("cluster", palette=c("red", "lightgrey", "blue"), title="Cluste
rs LISA") +
  tm_borders() +
  tm_layout(title = "Hotspots y Coldspots (Moran Local)")
```

```
##
## ── tmap v3 code detected ───────────────────────────────────────────────────
──────────
## [v3->v4] `tm_tm_fill()`: migrate the argument(s) related to the scale
of the
## visual variable `fill` namely 'palette' (rename to 'values') to fill.s
cale =
## tm_scale(<HERE>).[v3->v4] `tm_fill()`: migrate the argument(s) related
to the legend of the
## visual variable `fill` namely 'title' to 'fill.legend = tm_legend(<HER
E>)'[v3->v4] `tm_layout()`: use `tm_title()` instead of `tm_layout(title
= )`
```




```
# Fin del ejemplo
```