

# Mi proyecto

## Vecindad y autocorrelación espacial

### Proyecto Final de Analisis Espacial

#### Profesor :Jose Ramon Martinez Batlle

**Maria Magdalena Vilorio y Adalberto Guerrero P.**     *Estudiante, Universidad Autónoma de Santo Domingo (UASD)*

---

Mi resumen

*Keywords:* palabra clave 1, palabra clave 2

---

## 1 Introducción

‘A que cree que se debe la delincuencia en el país ¿A la falta de alternativas sanas (clubes, cine, teatro, etc.) para el entretenimiento?: Si’

## 2 Metodología

El objetivo de este proyecto es el análisis exploratorio de datos espaciales, funciones de homogeneidad espacial, tipos de vecindad, ponderadores y con el índice de autocorrelación espacial de Moran.

Usaremos como referencia la capa de provincias dominicanas y los resultados de la Encuesta Nacional de Hogares de Propósitos Múltiples de 2017 (ENHOGAR-2017, de. Te usaremos la pregunta de investigación, sobre la cual será realizado el análisis de autocorrelación al final.

...## Paquetes

- Carga el paquete `sf`, la colección `tidyverse` y los paquetes `spdep`, `lmtest`, `tmap` y `RColorBrewer`

```
library(sf)
```

```
## Linking to GEOS 3.7.1, GDAL 2.4.2, PROJ 5.2.0
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1      v purrr   0.3.3
## v tibble  2.1.3      v dplyr   0.8.3
## v tidyr   1.0.0      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
```

```
library(spdep)
```

```
## Loading required package: sp
```

```
## Loading required package: spData
```

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

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## as.Date, as.Date.numeric
```

```
library(tmap)
```

```
library(RColorBrewer)
```

## 2.1 Datos y unión

- Carga el conjunto de datos de “ENHOGAR 2017” (.csv), asignándolo al objeto en17. Nota que “ENHOGAR 2017” es una encuesta y, por lo tanto, recoge resultados referidos a una muestra, cuyo tamaño por provincia está recogido en la columna muestra. Carga también la capa geométrica (.gpkg) asignándola al objeto prov, y únelo a en17. Ambas fuentes se encuentran en la carpeta data. Recuerda el problema de la inconsistencia en el código entre ambas fuentes. Verifica consistencia luego de corregir el código (usa la práctica anterior como apoyo). Finalmente, une ambas fuentes, prov y en17.

```
en17 <- read.csv('material-de-apoyo-master/data/enhogar_2017.csv', check.names = F)
prov <- st_read(dsn = 'material-de-apoyo-master/data/divisionRD.gpkg', layer = 'PROVcenso2010')
```

```
## Reading layer 'PROVcenso2010' from data source '/home/franc/unidad-0-asignacion-99-mi-proyect
```

```
## Simple feature collection with 32 features and 4 fields
```

```
## geometry type: MULTIPOLYGON
```

```
## dimension: XY
```

```
## bbox: xmin: 182215.8 ymin: 1933532 xmax: 571365.3 ymax: 2205216
```

```
## epsg (SRID): 32619
```

```
## proj4string: +proj=utm +zone=19 +datum=WGS84 +units=m +no_defs
```

```
en17 <- en17 %>% mutate(ENLACE = ifelse(nchar(Código)==3, paste0('0', Código),Código))
match(en17$ENLACE, prov$ENLACE)
```

```
## [1] 9 18 25 13 24 28 6 14 19 20 5 15 26 27 2 17 21 31 3 4 10 16 7
## [24] 22 8 11 12 23 29 30 1 32
```

```
proven17 <- prov %>% inner_join(en17, by = 'ENLACE')
```

```
## Warning: Column 'ENLACE' joining factor and character vector, coercing into
## character vector
```

- Se Imprime en pantalla el sf resultante y genera un mapa que muestre tu pregunta para todo el país.

```
proven17 %>%
  dplyr::select(contains('A que cree que se debe la delincuencia en el país ¿A la falta de alter
  plot(breaks = 'jenks')
```



## 2.2 Conversión a *sp*

- Conviertiendo el objeto `proven17` a `SpatialPolygonsDataFrame` asignándolo a `proven17.sp`, mediante la función `as_Spatial`. Este paso es necesario para crear los objetos de vecindad. Verificamos que los nombres de columnas `proven17.sp` aparecen deformados (espacios sustituidos por puntos), y lo corregimos rescatando los nombres del objeto original `proven17`.

```
proven17.sp <- as_Spatial(proven17)
colnames(proven17.sp@data)[1:20]
```

```
## [1] "PROV"
## [2] "REG"
## [3] "TOPONIMIA"
```

```
## [4] "ENLACE"
## [5] "Código"
## [6] "Provincia"
## [7] "Principales.problemas.de.su.barrio.o.comunidad...La.falta.de.energía.eléctrica...Si"
## [8] "Principales.problemas.de.su.barrio.o.comunidad...La.falta.de.energía.eléctrica...No"
## [9] "Principales.problemas.de.su.barrio.o.comunidad...La.delincuencia...Si"
## [10] "Principales.problemas.de.su.barrio.o.comunidad...La.delincuencia...No"
## [11] "Principales.problemas.de.su.barrio.o.comunidad...El.desempleo...Si"
## [12] "Principales.problemas.de.su.barrio.o.comunidad...El.desempleo...No"
## [13] "Principales.problemas.de.su.barrio.o.comunidad...La.pobreza...Si"
## [14] "Principales.problemas.de.su.barrio.o.comunidad...La.pobreza...No"
## [15] "Principales.problemas.de.su.barrio.o.comunidad...El.consumo.de.drogas...Si"
## [16] "Principales.problemas.de.su.barrio.o.comunidad...El.consumo.de.drogas...No"
## [17] "Principales.problemas.de.su.barrio.o.comunidad...La.venta.de.drogas...Si"
## [18] "Principales.problemas.de.su.barrio.o.comunidad...La.venta.de.drogas...No"
## [19] "Principales.problemas.de.su.barrio.o.comunidad...El.costode.la.vida...Si"
## [20] "Principales.problemas.de.su.barrio.o.comunidad...El.costode.la.vida...No"
```

```
colnames(proven17.sp@data) <- proven17 %>% st_drop_geometry() %>% colnames
```

- Asignamos nombres de filas al objeto `proven17.sp` a partir de la columna `TOPONIMIA`.

```
row.names(proven17.sp) <- as.character(proven17.sp$TOPONIMIA)
```

### 2.3 Vecindad por contigüidad

- A partir de `proven17.sp`, creamos un objeto de vecindad por contigüidad, asignándolo a `proven17.nb`, usando criterio `queen`. Se Imprime en pantalla el resumen de dicho objeto de vecindad.

```
proven17.nb <- poly2nb(proven17.sp, queen=TRUE)
summary(proven17.nb)
```

```
## Neighbour list object:
## Number of regions: 32
## Number of nonzero links: 144
## Percentage nonzero weights: 14.0625
## Average number of links: 4.5
## Link number distribution:
##
##  1  2  3  4  5  6  7  8  9
##  1  2  5 11  5  4  2  1  1
## 1 least connected region:
## DISTRITO NACIONAL with 1 link
## 1 most connected region:
## LA VEGA with 9 links
```

- Evaluamos la cardinalidad, es decir, cuántos vecinos tiene cada geometría/elemento (que en este caso son provincias).

```
card(proven17.nb)
```

```
## [1] 1 6 4 4 3 7 4 4 6 5 2 3 9 3 4 2 3 4 3 4 5 7 5 4 6 6 4 5 8 4 5 4
```

- Imprimimos en pantalla la relación de vecinos de cada geometría.

```
sapply(proven17.nb, function(x) x)
```

```
## [[1]]
## [1] 32
##
## [[2]]
## [1] 3 4 13 17 22 31
##
## [[3]]
## [1] 2 4 10 22
##
## [[4]]
## [1] 2 3 10 16
##
## [[5]]
## [1] 7 15 26
##
## [[6]]
## [1] 9 13 14 19 20 24 29
##
## [[7]]
## [1] 5 10 22 26
##
## [[8]]
## [1] 11 12 23 30
##
## [[9]]
## [1] 6 13 14 18 19 25
##
## [[10]]
## [1] 3 4 7 16 22
##
## [[11]]
## [1] 8 12
##
## [[12]]
## [1] 8 11 23
##
```

```

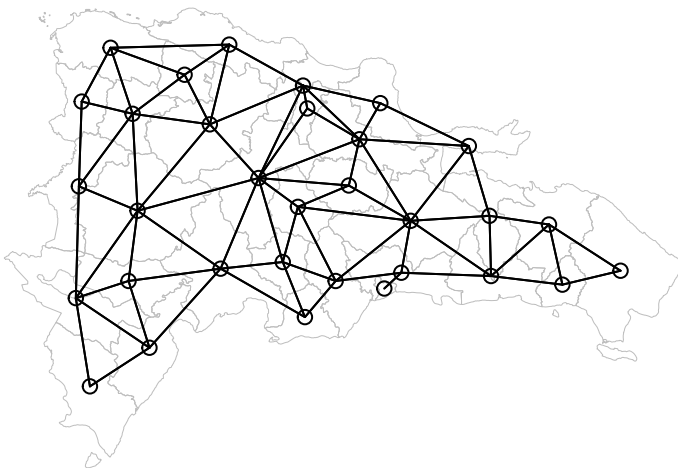
## [[13]]
## [1]  2  6  9 19 22 24 25 28 31
##
## [[14]]
## [1]  6  9 20
##
## [[15]]
## [1]  5 18 26 27
##
## [[16]]
## [1]  4 10
##
## [[17]]
## [1]  2 21 31
##
## [[18]]
## [1]  9 15 25 27
##
## [[19]]
## [1]  6  9 13
##
## [[20]]
## [1]  6 14 29 30
##
## [[21]]
## [1] 17 28 29 31 32
##
## [[22]]
## [1]  2  3  7 10 13 25 26
##
## [[23]]
## [1]  8 12 29 30 32
##
## [[24]]
## [1]  6 13 28 29
##
## [[25]]
## [1]  9 13 18 22 26 27
##
## [[26]]
## [1]  5  7 15 22 25 27
##
## [[27]]
## [1] 15 18 25 26
##
## [[28]]
## [1] 13 21 24 29 31
##

```

```
## [[29]]
## [1]  6 20 21 23 24 28 30 32
##
## [[30]]
## [1]  8 20 23 29
##
## [[31]]
## [1]  2 13 17 21 28
##
## [[32]]
## [1]  1 21 23 29
```

- Construimos un mapa de los vínculos de vecindad (grafo). Recuerdar que primero debemos de generar un mapa de las provincias, y luego se superpondrán los vínculos.

```
plot(proven17.sp, border="grey", lwd=0.5)
plot(proven17.nb, coordinates(proven17.sp), add=T)
```



\* Evaluamos si el objeto de vecindad es simétrico.

```
is.symmetric.nb(proven17.nb)
```

```
## [1] TRUE
```

## 2.4 Vecindad por número de vecinos

- A partir de `proven17.sp`, crearemos un objeto de vecindad por número de vecinos, en el que cada geometría tenga sólo un vecino, asignándolo a `proven17.nb.k1`. Imprime en pantalla el resumen de dicho objeto de vecindad. Recuerda crear un objeto de coordenadas de centroides, que en este ejercicio se sugiere con el nombre `coords`, y otro de identidades de cada geometría, para el cual se sugiere el nombre `ident`; ambos los usarás dentro de la función `knn2nb`. El resumen del objeto `proven17.nb.k1` debería mostrar 32 vínculos, el mismo número de regiones de `proven17.sp`, dado que cada región tiene un único vínculo.

```
coords <- coordinates(proven17.sp)
ident <- row.names(proven17.sp)
proven17.nb.k1 <- knn2nb(knearneigh(coords, k = 1), row.names = ident)
summary(proven17.nb.k1)
```

```
## Neighbour list object:
## Number of regions: 32
## Number of nonzero links: 32
## Percentage nonzero weights: 3.125
## Average number of links: 1
## Non-symmetric neighbours list
## Link number distribution:
##
## 1
## 32
## 32 least connected regions:
## DISTRITO NACIONAL AZUA BAORUCO BARAHONA DAJABÓN DUARTE ELÍAS PIÑA EL SEIBO ESPAILLAT INDEPENDENCIA
## 32 most connected regions:
## DISTRITO NACIONAL AZUA BAORUCO BARAHONA DAJABÓN DUARTE ELÍAS PIÑA EL SEIBO ESPAILLAT INDEPENDENCIA
```

- Evaluamos la cardinalidad, es decir, cuántos vecinos tiene cada geometría/elemento (que en este caso son provincias). Dado que se especificó anteriormente que sólo hubiese un único vecino, el vector debería ser una repetición de 1.

```
card(proven17.nb)
```

```
## [1] 1 6 4 4 3 7 4 4 6 5 2 3 9 3 4 2 3 4 3 4 5 7 5 4 6 6 4 5 8 4 5 4
```

- Imprime en pantalla la relación de vecinos de cada geometría.

```
sapply(proven17.nb, function(x) x)
```

```
## [[1]]
## [1] 32
##
## [[2]]
## [1] 3 4 13 17 22 31
##
## [[3]]
## [1] 2 4 10 22
##
## [[4]]
## [1] 2 3 10 16
##
## [[5]]
## [1] 7 15 26
##
```



```

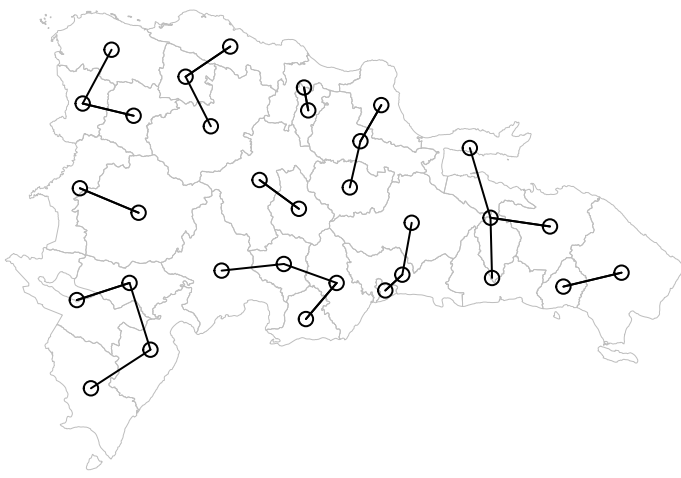
## [[6]]
## [1]  9 13 14 19 20 24 29
##
## [[7]]
## [1]  5 10 22 26
##
## [[8]]
## [1] 11 12 23 30
##
## [[9]]
## [1]  6 13 14 18 19 25
##
## [[10]]
## [1]  3  4  7 16 22
##
## [[11]]
## [1]  8 12
##
## [[12]]
## [1]  8 11 23
##
## [[13]]
## [1]  2  6  9 19 22 24 25 28 31
##
## [[14]]
## [1]  6  9 20
##
## [[15]]
## [1]  5 18 26 27
##
## [[16]]
## [1]  4 10
##
## [[17]]
## [1]  2 21 31
##
## [[18]]
## [1]  9 15 25 27
##
## [[19]]
## [1]  6  9 13
##
## [[20]]
## [1]  6 14 29 30
##
## [[21]]
## [1] 17 28 29 31 32
##

```

```
## [[22]]
## [1]  2  3  7 10 13 25 26
##
## [[23]]
## [1]  8 12 29 30 32
##
## [[24]]
## [1]  6 13 28 29
##
## [[25]]
## [1]  9 13 18 22 26 27
##
## [[26]]
## [1]  5  7 15 22 25 27
##
## [[27]]
## [1] 15 18 25 26
##
## [[28]]
## [1] 13 21 24 29 31
##
## [[29]]
## [1]  6 20 21 23 24 28 30 32
##
## [[30]]
## [1]  8 20 23 29
##
## [[31]]
## [1]  2 13 17 21 28
##
## [[32]]
## [1]  1 21 23 29
```

- Hacemos un mapa de los vínculos de vecindad (grafo). Recordar que primero debemos generar un mapa de las provincias (primera corrida de la función `plot`), y luego le superpondrás el mapa de los vínculos (segunda corrida de `plot`, con argumento `add=T`)

```
plot(proven17.sp, border="grey", lwd=0.5)
plot(proven17.nb.k1, coordinates(proven17.sp), add=T)
```



- Evaluamos si el objeto de vecindad es simétrico

```
is.symmetric.nb(proven17.nb)
```

```
## [1] TRUE
```

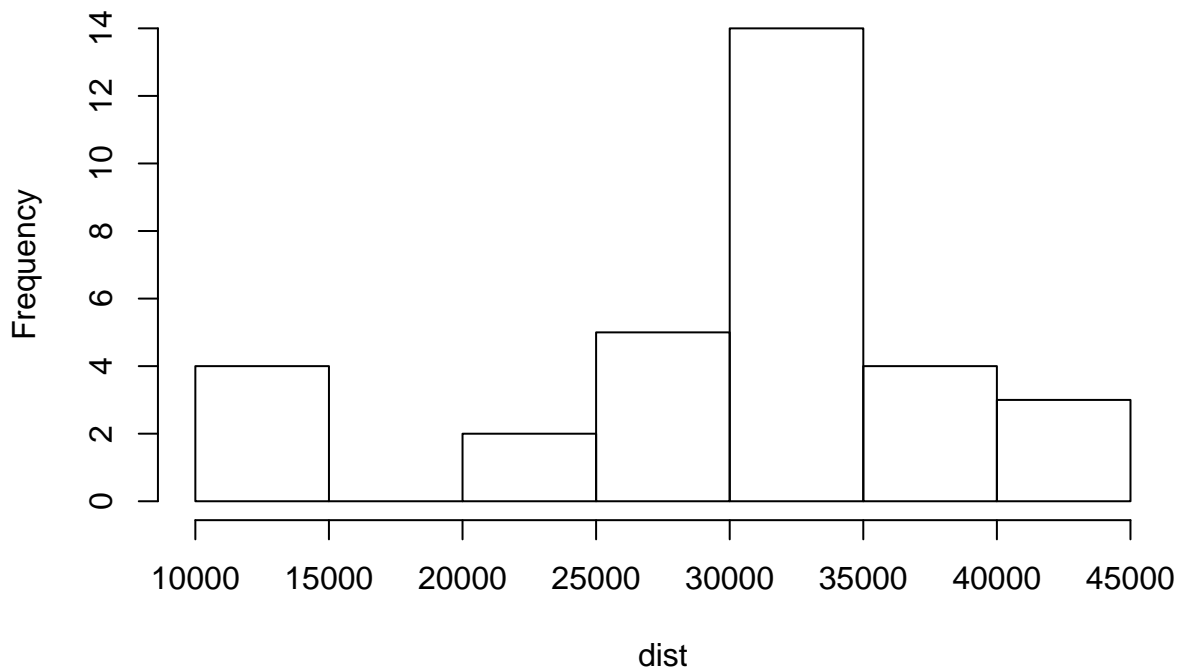
- Exploramos las distancias entre centroides de las geometrías a partir del objeto `proven17.nb.k1`. Para ello, creamos un objeto denominado `dist` donde se almacenan las distancias a partir de aplicar la función `nbdists` (recuerda que dentro de ésta debemos colocar el objeto `coords`). Esto Imprime en pantalla un resumen estadístico, y genera un histograma y un boxplot.

```
dist <- unlist(nbdists(proven17.nb.k1, coords))
summary(dist)
```

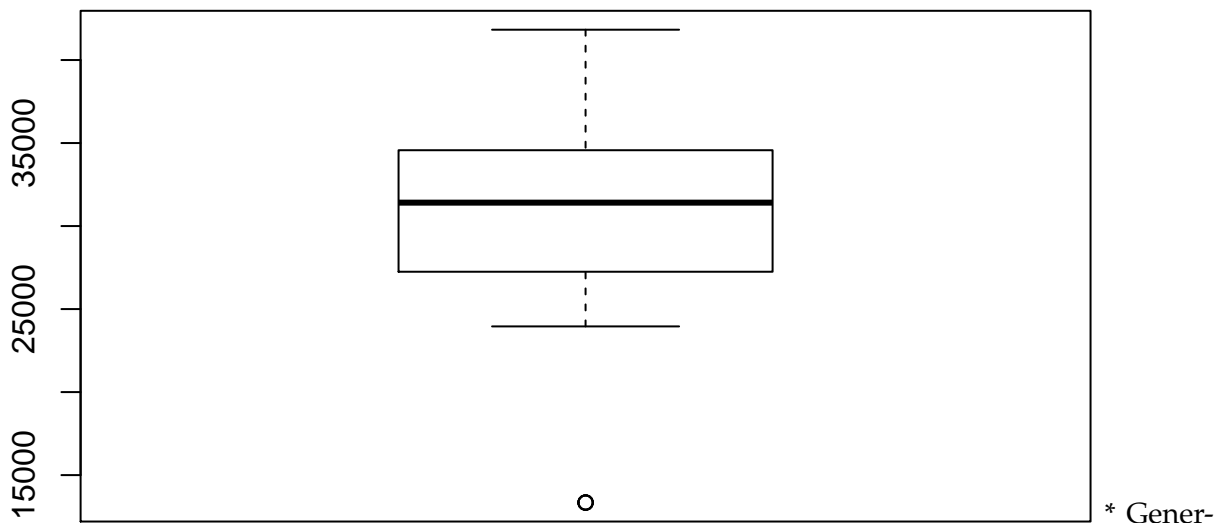
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  13343   27259   31411   29957   34552   41826
```

```
hist(dist)
```

### Histogram of dist



```
boxplot(dist)
```



aremos un objeto con la distancia mínima (objeto `distmin` usando la función `min`) y otro con la máxima (objeto `distmax` usando la función `max`). Luego, se determina qué posición ocupa en el vector `dist` ocupan los valores de `distmin` de `distmax`, y asígnalas a los objetos `indicemin` y `indicemax`, respectivamente. Posteriormente, utiliza dichas posiciones (`indicemin` y `indicemax`) dentro del índice de `ident` para determinar cuál o cuáles provincias se encuentran a la menor y a la mayor distancia en el conjunto del país.

```
(distmin <- min(dist))
```

```
## [1] 13343.28
```

```
(distmax <- max(dist))
```

```
## [1] 41826.21
```

```
indicemin <- which(dist==distmin)  
ident[indicemin]
```

```
## [1] "DISTRITO NACIONAL" "SANTO DOMINGO"
```

```
indicemax <- which(dist==distmax)  
ident[indicemax]
```

```
## [1] "SAMANÁ"
```

- Ordenaremos los nombres de provincias de menor a mayor distancia de separación con su vecino más próximo.

```
ident[order(dist)]
```

```
## [1] "DISTRITO NACIONAL" "SANTO DOMINGO"  
## [3] "ESPAILLAT" "HERMANAS MIRABAL"  
## [5] "DUARTE" "MARÍA TRINIDAD SÁNCHEZ"  
## [7] "PERAVIA" "SAN CRISTÓBAL"  
## [9] "SANCHEZ RAMÍREZ" "LA VEGA"  
## [11] "MONSEÑOR NOUEL" "DAJABÚN"  
## [13] "SANTIAGO RODRÍGUEZ" "MONTE PLATA"  
## [15] "PUERTO PLATA" "VALVERDE"  
## [17] "BAORUCO" "INDEPENDENCIA"  
## [19] "SANTIAGO" "SAN JOSÉ DE OCOA"  
## [21] "LA ALTAGRACIA" "LA ROMANA"  
## [23] "EL SEIBO" "HATO MAYOR"  
## [25] "SAN PEDRO DE MACORÍS" "MONTE CRISTI"  
## [27] "AZUA" "ELÍAS PIÑA"  
## [29] "SAN JUAN" "BARAHONA"  
## [31] "PEDERNALES" "SAMANÁ"
```

## 2.5 Ponderadores espaciales

- Generamos dos objetos de pesos espaciales a partir del objeto de vecindad por contigüidad; uno de ellos estandarizado por filas (asígnalo a `proven17.w.W`) y otro binario (asígnalo a `proven17.w.B`)

```
proven17.w.W <- nb2listw(proven17.nb)  
proven17.w.W
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 32
## Number of nonzero links: 144
## Percentage nonzero weights: 14.0625
## Average number of links: 4.5
##
## Weights style: W
## Weights constants summary:
##      n   nn S0      S1      S2
## W 32 1024 32 15.83669 132.7525
```

```
proven17.w.B <- nb2listw(proven17.nb, style = 'B')
proven17.w.B
```

```
## Characteristics of weights list object:
## Neighbour list object:
## Number of regions: 32
## Number of nonzero links: 144
## Percentage nonzero weights: 14.0625
## Average number of links: 4.5
##
## Weights style: B
## Weights constants summary:
##      n   nn S0  S1   S2
## B 32 1024 144 288 2968
```

## 2.6 Autocorrelación espacial de nuestra variable

Exploramos la autocorrelación espacial de nuestra variable utilizando el  $I$  de Moran global y el local.

- Usando tidyverse, generamos una columna de porcentaje de personas que respondió a nuestra pregunta respecto del tamaño de la muestra a nivel provincial (columna muestra). le Pondremos por nombre mivariable\_pct. Generamos una transformada a partir de la anterior, y le pondremos el nombre mivariable\_pct\_log. El objeto sf resultante se asignará a proven17\_mivar\_sf

```
mivariable <- 'A qué cree que se debe la delincuencia en el país: ¿A la falta de alternativas sa
proven17_mivar <- proven17 %>%
  st_centroid() %>%
  select(ENLACE, mivariable=contains(mivariable), muestra) %>%
  mutate('mivariable_pct' = mivariable/muestra*100,
         'mivariable_pct_log' = log1p(mivariable/muestra*100),
         x=unlist(map(geom,1)),
         y=unlist(map(geom,2))) %>%
```

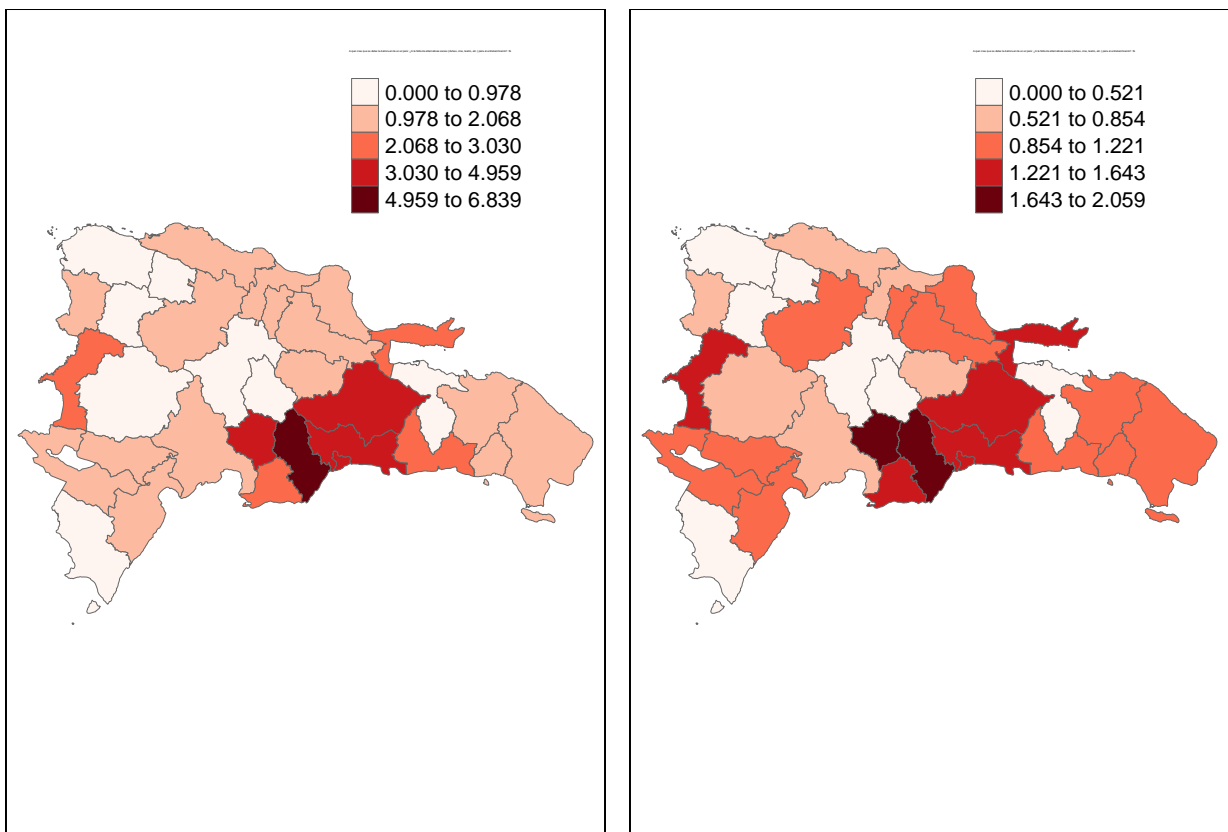
```
select(-muestra) %>%
st_drop_geometry()
```

```
## Warning in st_centroid.sf(.): st_centroid assumes attributes are constant
## over geometries of x
```

```
proven17_mivar_sf <- proven17 %>%
  inner_join(proven17_mivar, by = 'ENLACE') %>%
  dplyr::select(muestra, contains('mivvariable'), x, y, ENLACE, TOPONIMIA)
```

- Hacemos un mapa que muestre la variable, tanto en su versión original como transformada.

```
p1 <- tm_shape(proven17_mivar_sf) +
  tm_fill(col = "mivvariable_pct", style = 'jenks',
    palette = brewer.pal(9, name = 'Reds'), title = mivvariable) +
  tm_borders(lwd = 0.5)
p2 <- tm_shape(proven17_mivar_sf) +
  tm_fill(col = "mivvariable_pct_log", style = 'jenks',
    palette = brewer.pal(9, name = 'Reds'), midpoint = NA, title = mivvariable) +
  tm_borders(lwd = 0.5)
tmap_arrange(p1, p2)
```



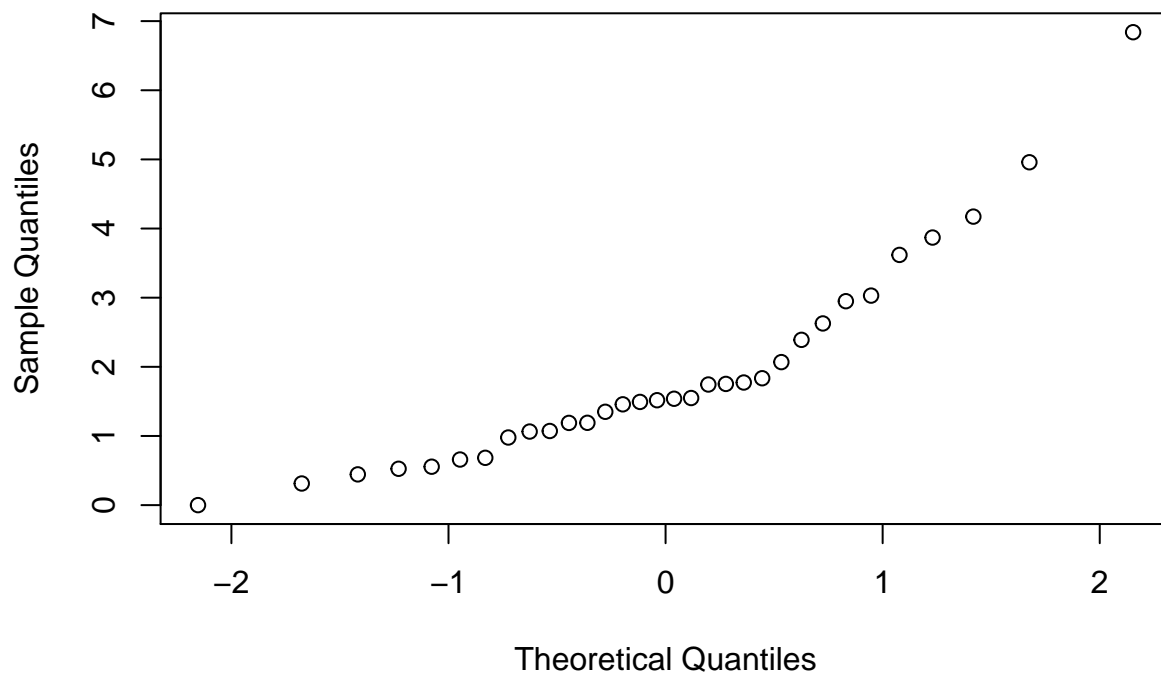
\* Compruebamos el supuesto de normalidad de nuestra variable, tanto en su versión original como transformada, mediante el gráfico cuantilar normal y la prueba de *Shapiro-Wilk*.

Tip: Como argumento de las funciones a continuación, usa la forma vectorial de tu variable original y transformada; e.g. `proven17_mivar_sf$mivariable_pct_log`

Tip: Si los puntos del gráfico cuantilar normal siguen una línea recta, y la prueba de Shapiro-Wilk resulta no significativa (es decir, el valor de  $p$  mayor que 0.05), entonces se asume como válido el supuesto de normalidad.

```
qqnorm(proven17_mivar_sf$mivariable_pct) #Versión original de la variable
```

### Normal Q-Q Plot



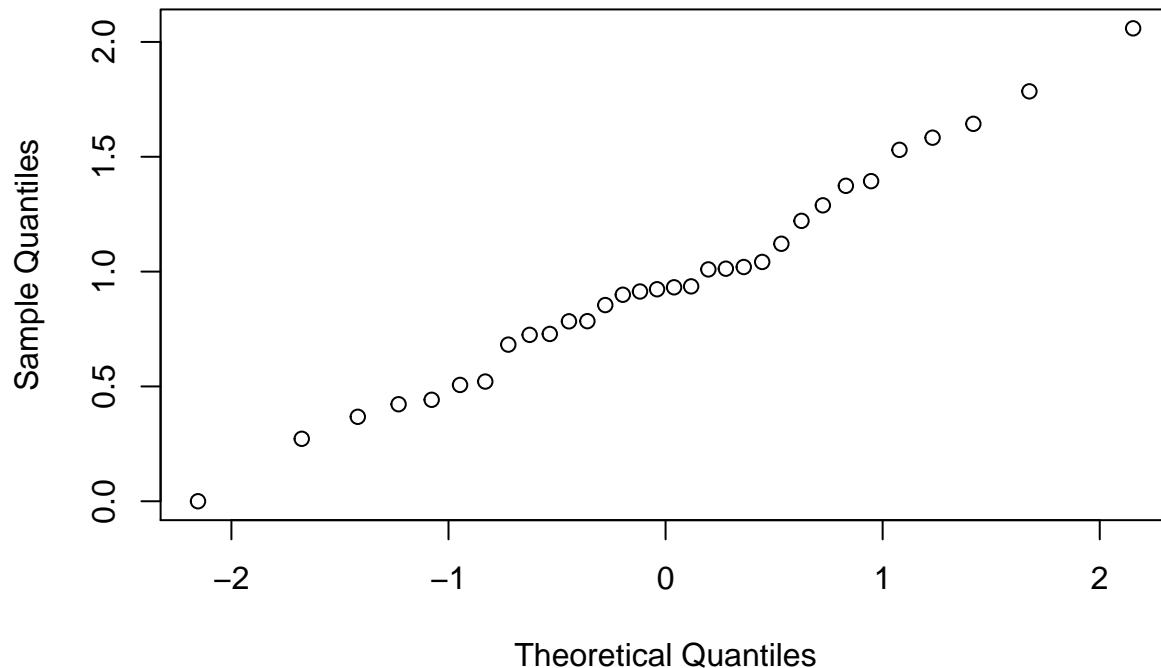
```
shapiro.test(proven17_mivar_sf$mivariable_pct) #Versión original de la variable
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  proven17_mivar_sf$mivariable_pct  
## W = 0.86785, p-value = 0.001036
```

```
qqnorm(proven17_mivar_sf$mivariable_pct_log) #Versión transformada de la variable (log)
```



### Normal Q-Q Plot



```
shapiro.test(proven17_mivar_sf$mivvariable_pct_log) #Versión transformada de la variable (log)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: proven17_mivar_sf$mivvariable_pct_log  
## W = 0.98533, p-value = 0.9311
```

### 3 Resultados

- Interpreta el resultado de la comprobación anterior aquí:
- 1-para la variable original=prueba de Shapiro-Wilk resulta significativa (es decir, el valor de  $p$  menor que 0.05), entonces se asume como no válido el supuesto de normalidad.
  - 2-para la variable modificada=prueba de Shapiro-Wilk resulta no significativa (es decir, el valor de  $p$  mayor que 0.05), entonces se asume como válido el supuesto de normalidad.
- Comprobamos el supuesto de homocedasticidad de tu variable respecto de  $x$  e  $y$ , tanto en su versión original como en la transformada, mediante la prueba de *Breusch-Pagan*.

Tip: Si el valor de  $p$  es menor que 0.05 (nivel de significancia convencional, aunque arbitrario), existe evidencia para rechazar la hipótesis de homocedasticidad.

```
proven17_mivar_sf %>% lm(mivvariable_pct ~ x, .) %>% bptest()
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: .  
## BP = 0.26322, df = 1, p-value = 0.6079
```

```
proven17_mivar_sf %>% lm(mivvariable_pct ~ y, .) %>% bptest()
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: .  
## BP = 3.8733, df = 1, p-value = 0.04906
```

```
proven17_mivar_sf %>% lm(mivvariable_pct_log ~ x, .) %>% bptest()
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: .  
## BP = 0.0025367, df = 1, p-value = 0.9598
```

```
proven17_mivar_sf %>% lm(mivvariable_pct_log ~ y, .) %>% bptest()
```

```
##  
## studentized Breusch-Pagan test  
##  
## data: .  
## BP = 5.4309, df = 1, p-value = 0.01978
```

- Interpreta el resultado de la comprobación anterior aquí: #1-para la variable original=el valor de  $p$  es mayor que 0.05 (nivel de significancia no convencional, aunque arbitrario), no existe evidencia para rechazar la hipótesis de homocedasticidad. #2-para la variable modificada=el valor de  $p$  es mayor que 0.05 (nivel de significancia no convencional, aunque arbitrario), no existe evidencia para rechazar la hipótesis de homocedasticidad.

En la eventualidad de que el supuesto normalidad y el de homocedasticidad no se cumplan, continúa con el procedimiento de estimar la autocorrelación la versión original o la transformada de tu variable, según elijas, aun cuando los resultados del análisis de autocorrelación espacial podrían no ser fiables.

## 6 Autocorrelación espacial global

- Comprobamos primero que existe consistencia en la secuencia de los nombres del objeto de vecindad y el *sf*.

```
match(attr(proven17.w.W$neighbours, "region.id"), proven17_mivar_sf$TOPONIMIA)==1:32
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [29] TRUE TRUE TRUE TRUE
```

- Aplicamos la prueba de autocorrelación espacial global para el *I* de Moran, usando los pesos estandarizados por filas como los binarios.

```
(gmoranz <- moran.test(x = proven17_mivar_sf$mivariable_pct_log, listw = proven17.w.W))
```

```
##
## Moran I test under randomisation
##
## data: proven17_mivar_sf$mivariable_pct_log
## weights: proven17.w.W
##
## Moran I statistic standard deviate = 2.2906, p-value = 0.01099
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.23185848      -0.03225806      0.01329463
```

```
(gmorana <- moran.test(x = proven17_mivar_sf$mivariable_pct_log, listw = proven17.w.B))
```

```
##
## Moran I test under randomisation
##
## data: proven17_mivar_sf$mivariable_pct_log
## weights: proven17.w.B
##
## Moran I statistic standard deviate = 2.119, p-value = 0.01705
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.19297774      -0.03225806      0.01129866
```

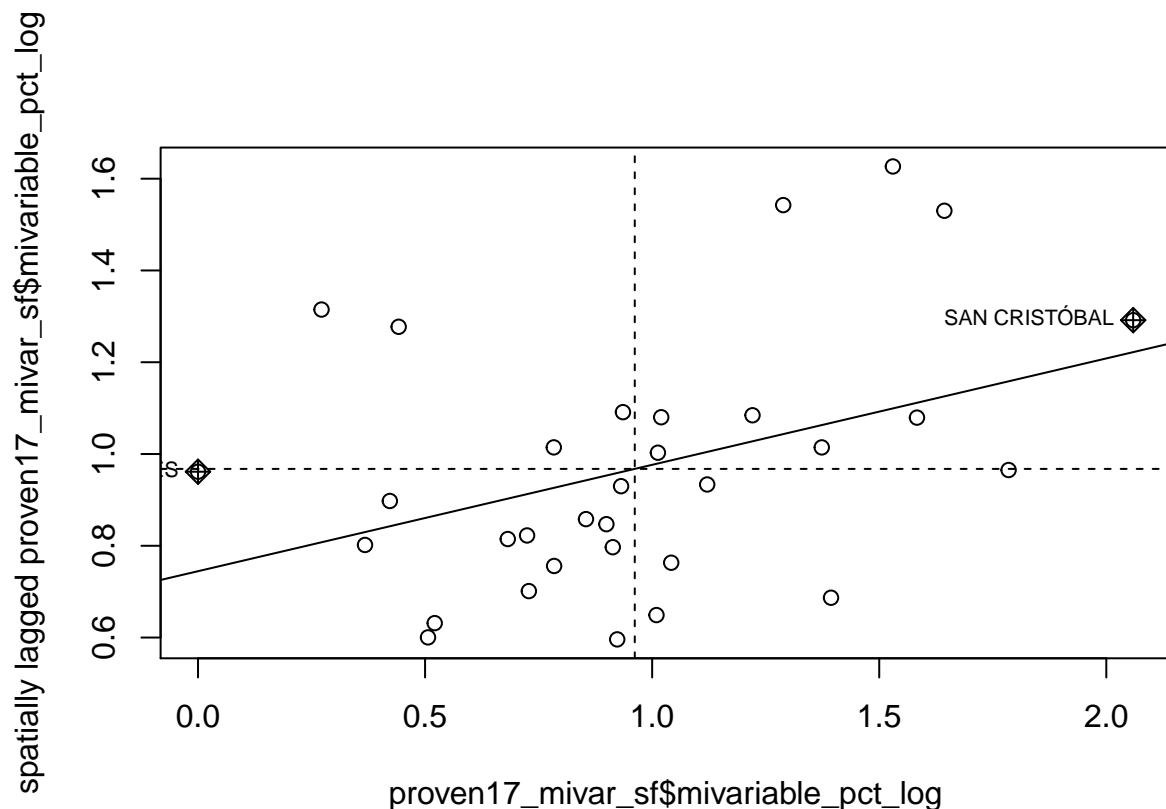
- Interpreta el resultado de la comprobación anterior aquí:

- 7 1-para los pesos estandarizados=el valor de  $p$  obtenido fue menor de 0.05, hay evidencia preliminar para rechazar la hipótesis nula de “no hay autocorrelación espacial”, y por lo tanto concluir que “sí hay autocorrelación espacial”
- 8 2-para los pesos binarios=el valor de  $p$  obtenido fue menor de 0.05, hay evidencia preliminar para rechazar la hipótesis nula de “no hay autocorrelación espacial”, y por lo tanto concluir que “sí hay autocorrelación espacial”

Tip: Si el valor de  $p$  obtenido fue menor de 0.05, hay evidencia preliminar para rechazar la hipótesis nula de “no hay autocorrelación espacial”, y por lo tanto concluir que “sí hay autocorrelación espacial”. En cualquier caso, es necesario que continúes evaluando la autocorrelación a nivel local en el siguiente paso, con independencia del resultado obtenido en la prueba global.

- Evaluamos la autocorrelación espacial local. Realizamos el diagrama de dispersión de Moran (*Moran scatterplot*), mediante la función `moran.plot`. Posteriormente, cargamos el script `lisaclusters.R`, y ejecutamos la función `lisamap` a nuestros datos para generar el mapa LISA. En la función `lisamap`, debemos introducir los siguientes argumentos: `objesp`, que es el objeto espacial denominado `proven17_mivar_sf`; `pesos`, que es el objeto de pesos, que será `proven17.w.W`.

```
moran.plot(x = proven17_mivar_sf$mivvariable_pct_log, listw = proven17.w.W)
```



```
source('lisaclusters.R')
lisamap(objesp = proven17_mivar_sf,
        var = 'mivvariable_pct_log',
```

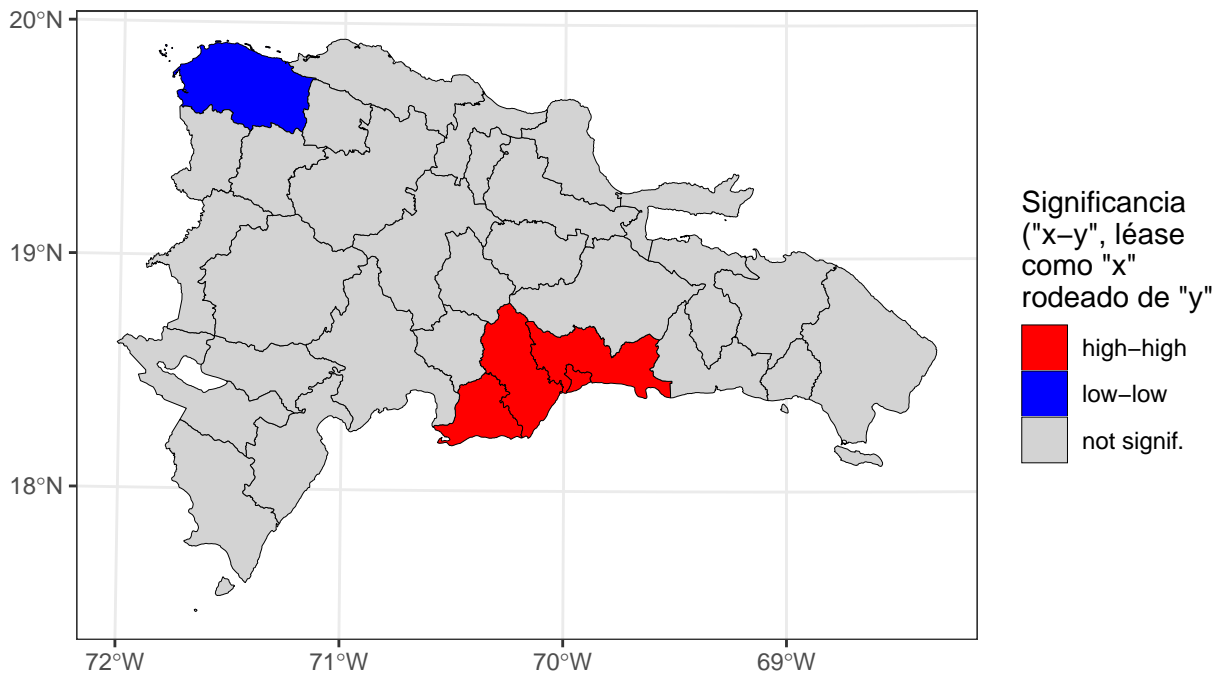
```

pesos = proven17.w.W,
tituloleyenda = 'Significancia\n("x-y", léase\ncomo "x"\nrodeado de "y"',
leyenda = T,
anchuratitulo = 1000,
tamanotitulo = 16,
fuentedatos = 'ENHOGAR 2017',
titulomapa = paste0('Clusters LISA de respuestas a la pregunta:\n"', mivariable, '"'))

```

```
## $grafico
```

## Clusters LISA de respuestas a la pregunta: "A qué cree que s



```

##
## $objeto
## Simple feature collection with 32 features and 11 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 182215.8 ymin: 1933532 xmax: 571365.3 ymax: 2205216
## epsg (SRID):    32619
## proj4string:     +proj=utm +zone=19 +datum=WGS84 +units=m +no_defs
## First 10 features:
##   muestra mivariable mivariable_pct mivariable_pct_log      x      y
## 1    4146         173      4.172697      1.6433941 400578.3 2044093
## 2     757           9      1.188904      0.7834008 306538.2 2055500
## 3     343           5      1.457726      0.8992365 253648.9 2048423
## 4     688          12      1.744186      1.0094845 265654.1 2010069
## 5     252           3      1.190476      0.7841190 226729.0 2151498

```

```

## 6      1198      21      1.752922      1.0126627 386306.5 2129851
## 7      198      6      3.030303      1.3938416 225124.2 2102815
## 8      325      5      1.538462      0.9315582 495201.2 2080870
## 9      815     11      1.349693      0.8542848 353886.4 2160683
## 10     201      3      1.492537      0.9133012 223355.0 2038578
##      ENLACE      TOPONIMIA      geom puntuacionz
## 1      1001 DISTRITO NACIONAL MULTIPOLYGON (((406845.9 20... 1.47572306
## 2      0502      AZUA MULTIPOLYGON (((322129.5 20... -0.38594119
## 3      0603      BAORUCO MULTIPOLYGON (((271940 2060... -0.13518662
## 4      0604      BARAHONA MULTIPOLYGON (((291856.5 20... 0.10347191
## 5      0405      DAJABÓN MULTIPOLYGON (((245433.3 21... -0.38438648
## 6      0306      DUARTE MULTIPOLYGON (((374434.8 21... 0.11035193
## 7      0707      ELÍAS PIÑA MULTIPOLYGON (((235630.8 21... 0.93550600
## 8      0808      EL SEIBO MULTIPOLYGON (((523436.4 20... -0.06521847
## 9      0109      ESPAILLAT MULTIPOLYGON (((385993.5 21... -0.23249553
## 10     0610      INDEPENDENCIA MULTIPOLYGON (((205698.2 20... -0.10474020
##      lagpuntuacionz      quad_sig
## 1      1.23048642      high-high
## 2      0.11418625      not signif.
## 3      -0.24811897      not signif.
## 4      -0.67691755      not signif.
## 5      -0.44526640      not signif.
## 6      0.08905067      not signif.
## 7      -0.59504145      not signif.
## 8      -0.06880244      not signif.
## 9      -0.22423955      not signif.
## 10     -0.35665546      not signif.

```

- Interpreta el resultado anterior aquí:

- 9 1-HAY un patron de relleno rojo y azul significa que existe autocorrelación local
- 10 2-HAY Un patrón de varias provincias coloreadas de rojo se atribuye a “efecto de contagio” importante. esto significa que hay autocorrelación espacial local
- 11 3-el las demas provincias se observa el gris significa que no hay autocorrelación espacial local

Tip: Si existe relleno rojo o azul, o ambos, significa que existe autocorrelación local. El relleno rojo (*hotspots*) significa que los valores altos del grupo coloreado son parecidos entre sí. Un patrón de varias provincias coloreadas de rojo se atribuye a “efecto de contagio” importante. Si sólo una provincia aparece en rojo, significa que las provincias de su entorno tienen valores parecidos, pero éstas últimas a su vez no tienen valores significativamente parecidos con su entorno ulterior (no se produce “contagio”). El relleno azul (*coldspots*) se interpreta de la misma manera que en el caso anterior, pero con los valores en este caso son bajos. Si sólo aparecen rellenos grises, significa que no hay autocorrelación local, y las provincias entonces presentan un patrón aleatorio de la variable analizada.

A modo de verificación, los mapas LISA de todas las variables asignadas se transcriben

a continuación. Nota que, de las variables asignadas, algunas no presentan autocorrelación espacial local.

## **12 Discusión o Conclusiones**

Atraves de este procedimientos pudimos verificar que la delincuencia en el pais se puede contagiar a varias provincias vecinas y que influye mucho en el crecimiento de la delincuencia la faltas de alternativas sanas como son clubes, cine , teatro etc. para el entretenimiento. atraves de la ejecucion del codigo pudimos darnos cuenta que el mismo tenia una distribucion normal en la variable modificada. mediante la prueba de Shapiro-Wilk, Breusch-Pagan y I de moran pudimos establecer los criterios para cada prueba descritos mas arriba. tambien pudimos demostrar que existe autocorrelacion espacial local y efecto de contagio importante.

...

## **13 Información de soporte**

Codigos, procedimientos de la clase de Vecindad y autocorrelacion espacial del profesor Jose Ramon Martinez Batlle.

...

## **14 *Script* reproducible**

...

## **15 Referencias**

Material de apoyo, suministrado por el profesor Jose Ramon Martinez Batlle. Capa de division de Provincia de La ONE. (Oficina Nacional de Estadisticas) Encuesta En hogar de la ONE.(Oficina Nacional de Estadisticas) Capa de ProvCenso2010 de la ONE.(Oficina Nacional de Estadisticas)