Guía laboratorio 2: Análisis uni-variados espaciales

Modulo 1: Índices de autocorrelación espacial

Carga de librerias

1. Cargue las librerías a utilizar (*terra*, *tmap*, *sf*, *spdep)*.

   library(terra)
   library(tmap)
   library(sf)
   library(spdep)

2. Descarge el shapefile llamado IncidenceMap, el cual tiene datos de incidencias por accidente ofídico a nivel departamental, defina su working directory y cargue los datos vectoriales.

   MapSB=vect("IncidenceMapClass/IncidenceMapClass.shp") #Leemos el mapa
   View(as.data.frame(MapSB)) #Vemos el mapa como un dataframe

3. Haga una gráfica de los datos de incidencia anual por cada 100.000 habitantes (Variable "InTot"). Para esto vamos a usar la librería *tmap* que hace mapas basado en el lenguaje por capas de ggplot. Cada + va aumentando una capa, y tenemos que usar el comando *st_as_sf()* sobre nuestro mapa para convertirlo a formato *sf* y que pueda ser usado por la librería. Con *tm_shape* definimos lo que graficaremos y la base de datos shapefile a utilizar, con *tm_polygons* elegimos las características de los polígonos y como los vamos a colorear, y con *tm_legend* definimos la leyenda de la figura.

   tm_shape(st_as_sf(MapSB))+tm_polygons(style="quantile",col="InTot")+
   tm_legend(outside=TRUE,text.size=0.8) #Grafico Mapa Incidencia

4. Antes de calcular los indices, necesitamos definir los vecinos. Primero vamos a trabajar con vecinos **contiguos**, pero hay mas maneras de declarar vecinos. Para declarar los vecinos contiguos usamos el comando *poly2nb()*. En este comando toca transformar nuestro mapa nuevamente a *sf*.

   nb = poly2nb(st_as_sf(MapSB),row.names=MapSB$ID,queen=TRUE) #Calculamos vecinos CONTIGUOS

5. Ahora, vamos a calcular la lista de pesos de cada vecino, lo cual es algo que hay que hacer para cada tipo de análisis de autocorrelación espacial, ya que gracias a esto podemos calcular los estadísticos. Usaremos la funci'on *nb2listw()* para definir estos pesos, y en esta función *style="U"* define un peso igual para cada vecino, donde la suma total da igual a 1.

   lw = nb2listw(nb,style="U")#Asignamos los pesos a cada vecino

6. Con estos pesos ya podemos calcular la incidencia desfazada (con lag), usando el comando *lag.litw()*. Para esto usamos nuestra lista de pesos, y la variable *InTot*. Una ves calculada esta incidencia desfazada, podemos hacer el moran plot para ver si tenemos algún tipo de asociación, y mirar una regresión lineal normal para ver el comportamiento y el valor de la pendiente.

```
Inc.lag=lag.listw(lw,MapSB$InTot) #Caluclo incidencia laggeada
plot(Inc.lag~MapSB$InTot) #Grafico Moran
summary(lm(Inc.lag~MapSB$InTot)) #Regresion lineal entre datos "lageados" y datos
```

7. Ahora, vamos a calcular el p-val de forma corregida y analítica, y también con simulaciones de Montecarlo. Recuerde que siempre es mejor usar las simulaciones de Montecarlo para evitar tener tantos supuestos. El comando *moran.test()* nos permite calcular de forma analítica el p-value y el índice de moran, mientras que *moran.mc()* nos permite hacer simulaciones de montercarlo para calcularlo.

```
moran.test(MapSB$InTot,lw) #p-val analítico con suposiciones
#P-VAL CON MONTECARLO
MC=moran.mc(MapSB$InTot,lw,nsim=100000) #P val con menos suposiciones. Ojo con nsim, ya que es el que define que tan robusto es mi estimación de la distribución.
plot(MC)#Grafico distribucion p-val a partir de montecarlo
MC
```

8. Tenemos una autocorrelación espacial en nuestros datos de incidencia por mordedura de serpientes. Ahora, queremos cambiar el tipo de vecinos y usar K-Nearest. Para esto primero debemos sacar un data.frame que tenga las coordenadas de los centroides de cada departamento, para sobre esto calcular los vecinos mas cercanos con base en distancias entre puntos. Con *crds()* obtenemos un data.frame de coordenadas de un set de puntos, y con *centroids()* obtenemos los centroides de un mapa vectorial. Con *knearneigh()* definimos la lista de vecinos mas cercanos, pero sale en un formato que no se puede usar para calcular los pesos con *nb2listw()*. Por esto, toca usar el comando *knn2nb()* para pasar el formato al necesario y poder realizar nuestro análisis.

```
coo=crds(centroids(MapSB)) #Obtenemos centroides
nb2=knearneigh(coo,k=3) #3 vecinos mas cercanos
nb2=knn2nb(nb2) #Pasamos a formato de vecinos
```

9. Ya con esto, podemos volver a calcular los pesos y calcular el índice de Moran.

```
#PESOS
lw=nb2listw(nb2,style="U",zero.policy=T)
MI = moran.mc(MapSB$TotCases,lw,nsim=10000,zero.policy=T)
MI
```

10. Seguimos obteniendo una autocorrelación espacial positiva para la incidencia por cada 100.000 habitantes por año de accidente ofídico. Ahora vamos a calcular el valor de Moran local. Para esto, el procedimiento es el mismo. En este caso vamos a trabajar con los 4 vecinos mas cercanos.

    coo=crds(centroids(MapSB)) #Definimos nuevamente los centroides y sacamos las coordenadas
    nb2=knearneigh(coo,k=4) #Usamos los 4 vecinos mas cercanos
    nb2=knn2nb(nb2) #Cambiamos el formato

11. Algo interesante de analizar es ver un plot en el que tengamos las conexiones obtenidas entre cada departamento y sus vecinos, y esto lo podemos graficar con los gráficos default de R.

    plot(MapSB,border="lightgrey") #Graficamos los bordes de los departamentos
    plot(nb2,coo,pch=19,cex=0.6,add=TRUE,col="red") #Añadimos las líneas en rojo de los vecinos.

12. Antes de calcular el moran local, tenemos que calcular la lista de pesos.

    wb=nb2listw(nb,zero.policy=T,style="U")

13. Ya con esto podemos calcular el moran local. Usamos el comando *localmoran_perm()* ya que vamos a calcular p-values por permutaciones.

    #MORAN I LOCAL
    local_moran=localmoran_perm(MapSB$InTot,wb,nsim=1000)

14. La variable *local_moran* consiste en una matriz con 9 columnas, donde *Ii* tiene el valor del índice para cada departamento, y *Pr(z != E(Ii))* es el p-value de las simulaciones por Montecarlo. Vamos a adicionar el índice de moran y el pval a nuestro mapa, y lo haremos con base en la posición de la columna.

    #MAPA MORAN LOCAL
    MapSB$LocMor=local_moran[,1] #Local moran
    MapSB$LocMorPVAL=local_moran[,6] #pval

15. Por último, vamos a graficar únicamente el valor del moran local para los lugares con significancia, dejando los sitios no significativos un valor de cero. Para esto primero creamos una variable que es significancia que va a ser *TRUE* o *FALSE* con base en nuestro p-value, y luego multiplicamos el índice de moran por esta variable.

    MapSB$sig=(MapSB$LocMorPVAL<0.05) #Significancia: 0.05, por ende a los menores les damos valor de TRUE (1) y los mayores de FALSE (0)

MapSB$LocMorsig=MapSB$LocMor*MapSB$sig #Aca hacemos la multiplicación, todo lo significativo (MapSB$sig < 0.05) es multiplicado por 1, y lo otro por cero.

16. Ahora graficamos esta variable. Podemos observar que la parte mas afectada siendo valores positivos de autocorrelación aquellos donde tenemos mas casos en cada departamento predicen mas casos en sus vecinos.

```
tm_shape(st_as_sf(MapSB))+tm_polygons(col="LocMorsig",palette="-RdBu")+
  tm_legend(outside=TRUE,text.size=0.8)
```

17. Por último, vamos a calcular el índice de GetisOrd con base en la misma lista de pesos que usamos para el moran local. Usamos el comando *localG_perm()* para obtener los resultados con permutaciones y calcular p-values sin suposiciones.

```
Gi=localG_perm(MapSB$InTot,wb,nsim=1000)
```

18. Primero añadimos al mapa los valores del índice, y usamos *as.matrix()* para pasar a formato numerico:

```
Giv=matrix(Gi) #Formato vector para añadir al mapa
```

19. El output de *local_perm()* viene también con atributos de clase, los cuales tienen mas información como el tipo de cluster y los p-values. Para obtener el tipo de cluster (High, Low) usamos el atributo cluster:

```
Clust=attr(Gi,"cluster") #Obtenemos el numero de cluster
```

20. Para obtener los pvalues, tenemos que guardar como data.frame el atributo internals de nuestro Gi:

```
Res=data.frame(attr(Gi,"internals"))
```

21. Ya con esto añadimos los datos al mapa:

```
Pval=Res$Pr.z....E.Gi...Sim #Obtenemos los p-values con simulaciones
Pval=Pval<0.05 #Definimos significativos y no significativos
MapSB$Gi=Giv #Añadimos el Getis Ord al mapa
MapSB$Giclust=Clust #Añadimos el tipo de cluster
MapSB$GiPval=MapSB$Gi*Pval #Añadimos solo los Getis Ord significativos
```

22. Ya con esto, procedemos a gráficar los Getis-ord significativos.

```
tm_shape(st_as_sf(MapSB))+tm_polygons(col="GiPval",style="pretty",palette="-RdBu")
+
```

tm_legend(outside=TRUE,text.size=0.8)

Podemos observar que la región que forma un clúster de mayor riesgo esta en la parte amazónica del país.

Preguntas laboratorio:

1. Defina que significan los *style* de *"W"* , *"B"*, *"C"*, *"minmax"* y *"S"* y como cambian los pesos según cada uno de estos estilos.
2. Calcule el índice global de moran usando vecinos mas cercanos de 3,4,5, y 6. Haga un gráfico que en el eje Y este el índice global y en el x el numero de vecinos más cercanos utilizado.
3. Calcule el índice de Getis-Ord usando vecinos adyacentes y compare con los obtenidos con lo obtenido usando los 4 vecinos mas cercanos en este laboratorio.