Laboratorio 5

Mariana Cubero Corella

7/11/2020

Laboratorio 5 Autocorrelación Espacial

Introducción

En algunos casos la información solo está disponible agregada para áreas, como provincias, cantones, UGMs u otras. Cuando surge la interrogante sobre la asociación que pueda existir de un fenómenos en dos o más áreas de estudio. Similar al caso de series temporales, la autocorrelación espacial mide la relación de una misma variable consigo misma pero en dos ubicaciones distintas. Estos se conoce como autocorrelación espacial. Existen diferentes abordajes para estudiar estas autocorrelaciones espaciales, una de ella es el I de Moran. Si el I de Moran, que mide la autocorrelación, es cercano a 1 indica que hay similitudes identificadas en las observaciones(agrupados, hot spots), si es cercano a -1 quiere decir que no hay similitudes entre valores cercanos (dispersos).

Este ejercicio trata de una explicación sobre la construcción de la autocorrelación espacial.

Respuesta

1.

Explique las primeras 5 líneas devueltas por el comando str(w)

El significado de las primeras 5 líneas corresponde a los polígonos adyacentes con respecto a un polígono especifico, del 1:3 se tienen 3 polígonos adyacentes o 3 vecinos del polígono 1, de 1:4 tiene 4 polígonos adyacentes o vecinos que corresponden al polígono 2, de 1:2 tiene 2 polígonos adyacentes que corresponden a los vecinos del polígono 3, de 1:2 corresponden a los vecinos del polígono 4 y de 1:3 se tienen 3 polígonos adyacentes o vecinos del polígono 5.

2.

¿Cómo interpreta los resultados de las pruebas de significancia?

Prueba si hay patrones de correlacion de agrupaciones donde H0 es que están distribuidos aleatoriamente en el espacio: ### Moran Test

H0: No hay autocorrelación espacial H1: Hay autocorrelación espacial.

```
moran.test(p$value, ww, randomisation = F)
```

```
##
##  Moran I test under normality
##
## data:  p$value
## weights: ww
##
## Moran I statistic standard deviate = 2.3372, p-value = 0.009714
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##         0.1728896        -0.2500000         0.0327381
```

Con un alpha 5% y un p-value 0.009 se tiene suficiente evidencia estadística para rechazar H0, con lo que se está en presencia de autocorrelación espacial.

Moran test (con simulación de Monte Carlo)

H0: No hay autocorrelación espacial H1: Hay autocorrelación espacial.

```
moran.mc(p$value, ww, nsim = 99)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 100
##
## statistic = 0.17289, observed rank = 98, p-value = 0.02
## alternative hypothesis: greater
```

Con un alpha de 5% y un p-value de 0,02 se tiene suficiente evidencia estadística para rechazar H0, con lo que se está en presencia de autocorrelación espacial.

3.

¿Cuál es el valor máximo que podemos usar en nsim?

Da problemas con un nsim =1100, se procede a realizar una busqueda del máximo valor de nsim antes de generar error por tamaño del valor en nsim.

Se determina que nsim = 120 es el máximo valor en la prueba antes de generar problemas por tamaño.

```
moran.mc(p$value, ww, nsim = 120)
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 121
```

```
##
## statistic = 0.17289, observed rank = 119.5, p-value = 0.0124
## alternative hypothesis: greater
```

4.

Muestre como usar la función de "Greary" para calcular la C de Greary?

```
n <- length(p)
x <- p$value
xbar <- mean(x)

#Greary

# Parte del denominador (segunda parte de la formula)
dx <- (x-xbar)
# Parte del numerador (segunda parte de la formula)
xi <- rep(x, each = n)
xj <- rep(x)
xixj <- xi-xj

# Matriz w_ij
wm <- nb2mat(w, style = 'B')

# Sumatoria de la multiplicación de W_ij*(xi-xj)^2
spwm <- sum(wm*(xixj)^2)

# Sumatoria de denominador de la segunda expresión de la formula
smw <- (2*sum(wm))

# Segunda expresión de la formula
sw <- spwm / smw

vr <- ((n-1)/sum((dx)^2))

CG <- vr* sw

CG

## [1] 0.5357143
```

En la C de Greary,el numerador es un factor de normalización, establece las diferencias entre la unidad de análisis y sus vecinos, el valor del coeficiente es entre los mismos vecinos y no contra la media global como en el I de Moran. Esta de C de Greary cuando tiene valores grandes sugieren correlación negativa, de igual forma cuando la C tiene valores bajos sugiere correlación positiva.

5.

Escriba su propia prueba de simulación de Monte Carlo para calcular los valores p para la I de Moran, replicando los resultados que obtuvimos con la función de spdep. Muestre un histograma de los valores simulados.

```
set.seed(1234)
bperm <- moran.mc(y, listw=ww, nsim=99)
bperm
```

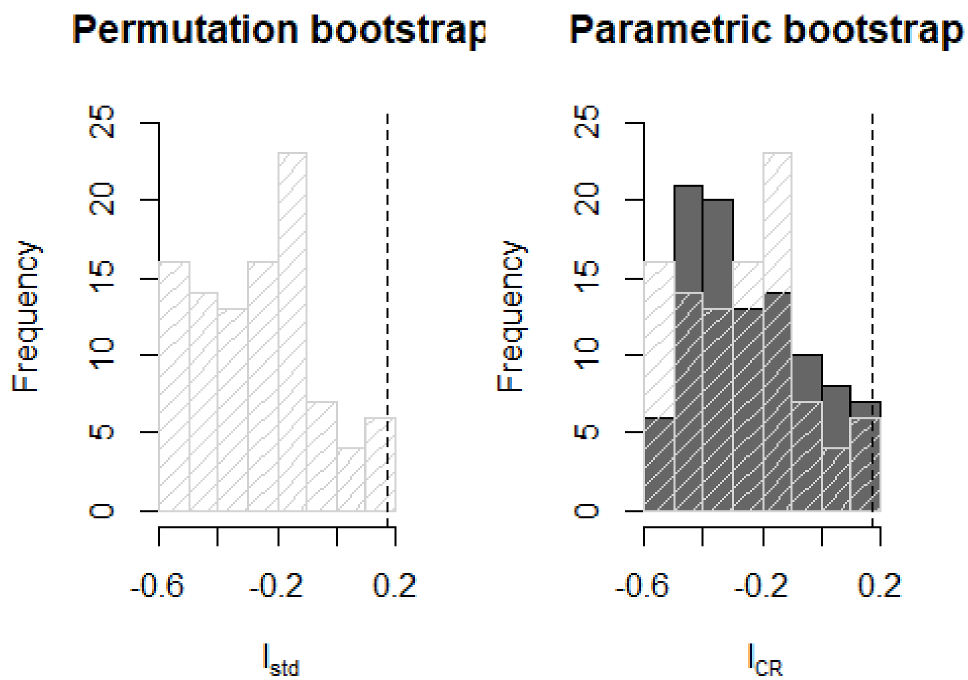
```
##
##  Monte-Carlo simulation of Moran I
##
## data:  y
## weights: ww
## number of simulations + 1: 100
##
## statistic = 0.17289, observed rank = 98, p-value = 0.02
## alternative hypothesis: greater
```

```
y <- p$value
ybar <- mean(y)
CR <- function(var, mle) rpois(length(var), lambda=mle) # genera poissons
MoranI.pboot <- function(var, i, listw, n, S0, ...) {
  return(moran(x=var, listw=listw, n=n, S0=S0)$I)
}
set.seed(1234)
library(boot)

boot2 <- boot(y, statistic=MoranI.pboot, R=99, sim="parametric",
  ran.gen=CR, listw=ww, n=length(y), S0=Szero(ww), mle=ybar)
pnorm((boot2$t0 - mean(boot2$t))/sd(boot2$t[,1]), lower.tail=FALSE)
```

```
## [1] 0.02015095
```

```
oopar <- par(mfrow=c(1,2))
xlim <- range(c(bperm$res, boot2$t[,1]))
hist(bperm$res[-length(bperm$res)], main="Permutation bootstrap",
xlab=expression(I[std]), xlim=xlim, density=15, angle=45, ylim=c(0,25))
abline(v=bperm$statistic, lty=2)
hist(boot2$t, col=rgb(0.4,0.4,0.4), main="Parametric bootstrap",
xlab=expression(I[CR]), xlim=xlim, ylim=c(0,25))
hist(bperm$res[-length(bperm$res)], density=15, angle=45, add=TRUE)
abline(v=boot2$t0, lty=2)
```



6.

```
#Función a completar
# gearyC <- ((n-1)/sum(( "----")\^2)) * sum(wm * (" --- ")\^2) / (2 *
sum(wm))
n <- length(p)
x <- p$value
xbar <- mean(x)
dx <- (x-xbar)
# Parte del numerador (segunda parte de la formula)
xi <- rep(x, each = n)
xj <- rep(x)
xixj <- xi-xj
wm <- nb2mat(w, style = 'B')

gearyC <- ((n-1)/sum((dx)^2))*sum(wm*(xixj)^2)/(2*sum(wm))
gearyC

## [1] 0.5357143

#Prueba
ww <-  nb2listw(w, style='B')
geary.test(dx, ww, randomisation = TRUE, zero.policy = NULL,
           alternative = "greater", spChk = NULL, adjust.n = TRUE)

##
##  Geary C test under randomisation
##
```

```
## data:  dx
## weights: ww
##
## Geary C statistic standard deviate = 2.42, p-value = 0.007761
## alternative hypothesis: Expectation greater than statistic
## sample estimates:
## Geary C statistic       Expectation          Variance
##        0.53571429        1.00000000        0.03680884
```

Se tiene el mismo resultado para los dos casos.