Laboratorio 5

Katherine Barquero

29/10/2020

Introducción

El objetivo de este informe es presentar los pasos para realizar el proceso de estimación de la autocorrelación espacial, primero se inicia con la comprensión de la autocorrelación temporal y luego se estima la espacial a través del estadístico I'Moran, pero además muestra el proceso de estimación de otras técnicas como la estimación a partir de la C de Greary y la simulación de Monte Carlo.

Los datos utilizados no se describen de manera exhaustiva pero responden a locaciones de comunas de Luxemburgo y pretende identificar a partir de los métodos mencionados cuál es el grado en el que dos de estas ubicaciones espaciales son similares entre sí, a partir de la estimación de la autocorrelación espacial.

Question 1: Explain the meaning of the first 5 lines returned by str(w)

El objeto *w* se obtiene a partir de la función spdep la cuál permite determinar cuáles polígonos están cerca uno de otros a partir del criterio de adyacencia. Por lo tanto, las 5 primeras líneas presentan los identificadores de las adyancencias estimadas para los polígonos bajo análisis. Por ejemplo, en el caso de la primera línea 1:3 indica que se tienen 3 polígonos adyacentes o vecinos del polígino 1, para la línea 1:4 se idnica que se tienen 4 polígonos vecinos con respecto al polígono 2, y así sucesivamente.

```
## List of 5
##  $ : int [1:3] 2 4 5
##  $ : int [1:4] 1 3 4 5
##  $ : int [1:2] 2 5
##  $ : int [1:2] 1 2
##  $ : int [1:3] 1 2 3
##  - attr(*, "class")= chr "nb"
##  - attr(*, "region.id")= chr [1:5] "0" "1" "2" "3" ...
##  - attr(*, "call")= language poly2nb(pl = p, row.names = p$Id)
##  - attr(*, "type")= chr "queen"
##  - attr(*, "sym")= logi TRUE
```

Question 2: How do you interpret these results (the significance tests)?

La I Moran analiza la hipótesis nula que indica que los datos están distribuidos aleatoriamente o bien que no hay autocorrelación espacial. Si se utiliza un nivel de significancia del 5%, tal y como se observa a continuación el valor del p-value obtenido es de 0.009714, por lo que hay evidencia estadística para no aceptar la hipótesis, es decir, hay evidencia de que los datos presentan autocorrelación espacial.

```
##
##  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
```

Question 3: What is the maximum value we can use for nsim?

Se realizan varias pruebas utilizando distintos valores para nsim, los resultados obtenidos se presentan a continación. Con estos datos el máximo valor que puede usarse es 120, a partir de allí se generar un error que indica que el valor del número de permutaciones es muy alto para este set de datos.

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 100
##
## statistic = 0.17289, observed rank = 98.5, p-value = 0.015
## alternative hypothesis: greater
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 101
##
## statistic = 0.17289, observed rank = 101, p-value = 0.009901
## alternative hypothesis: greater
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 106
##
## statistic = 0.17289, observed rank = 105, p-value = 0.009434
## alternative hypothesis: greater
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 111
##
## statistic = 0.17289, observed rank = 110, p-value = 0.009009
## alternative hypothesis: greater
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 116
##
## statistic = 0.17289, observed rank = 115.5, p-value = 0.008621
## alternative hypothesis: greater
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  p$value
## weights: ww
## number of simulations + 1: 121
##
## statistic = 0.17289, observed rank = 120.5, p-value = 0.008264
## alternative hypothesis: greater
```

Question 4: Show how to use the ‘geary’ function to compute Geary’s C

La fórmula de Geary’s C viene dada por la siguiente expresión

$$C = \frac{(n-1)}{2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(x_i - x_j)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

En esta expresión, el primer término hace referencia a un factor de normalización, mientras que en el segundo se obtienen el numerador las diferencias entre la unidad de análisis y sus vecinos. En este caso, contrario a lo que indica la I Moran, las diferencias (en la parte del numerador del segundo término) no se realizan con respecto a la media, si no que se utiliza cada valor del vecino o adyacencia estimada.

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

#Compute Geary's C

# Estimación del numerador (expresión derecha de la fórmula)
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)


# Estimación del denominador(expresión derecha de la fórmula)
dx <- (x-xbar)


# Denominador, parte izquierda de la fórmula
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
```

```
# La siguiente función permite estimar la C de Gary, los resultados son sumamente similares a los anteriorem
ente indicados "a pie"

col.W <- nb2listw(w, style="W")
str(geary(p$value, col.W, length(w), length(w)-1,
 Szero(col.W)))
```

```
## List of 2
##  $ C: num 0.522
##  $ K: num 1.43
```

Question 5: Write your own Monte Carlo simulation test to compute p-values for Moran's I, replicating the results we obtained with the function from spdep. Show a histogram of the simulated values.

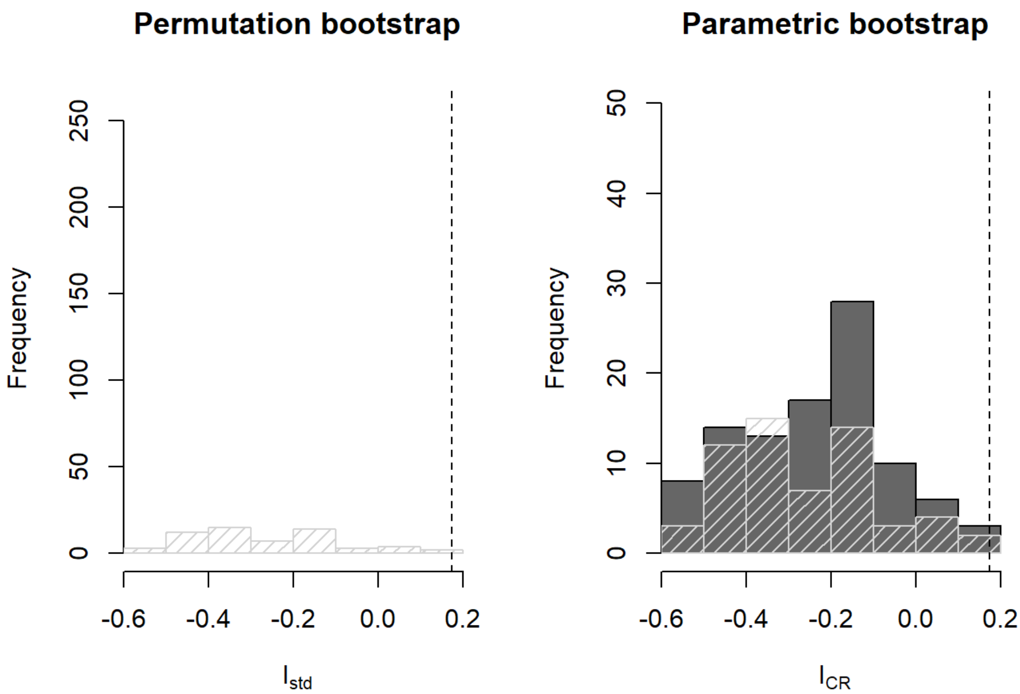
```
set.seed(12345)
MC <- moran.mc(y, listw=ww, nsim=60)
MC
```

```
##
##  Monte-Carlo simulation of Moran I
##
## data:  y
## weights: ww
## number of simulations + 1: 61
##
## statistic = 0.17289, observed rank = 60.5, p-value = 0.01639
## alternative hypothesis: greater
```

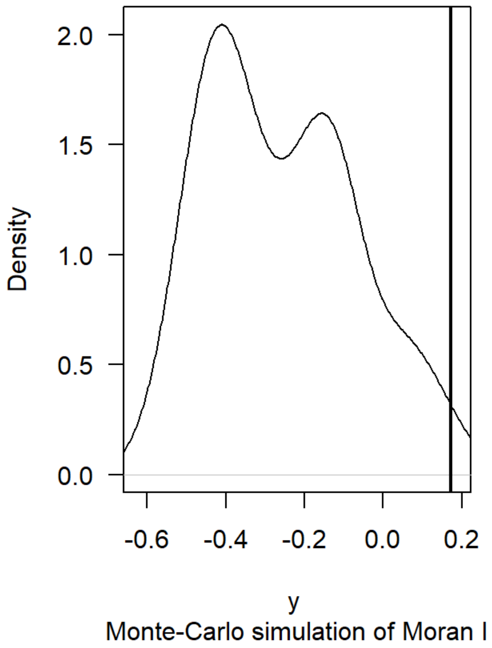
```r
y <- p$value
ybar <- mean(y)
CR <- function(var, mle) rpois(length(var), lambda=mle) # genera poissons
MoranI.pboot <- function(var, i, listw, n, S0, ...) {
  return(moran(x=var, listw=listw, n=n, S0=S0)$I)
}

set.seed(12345)


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

```
## [1] 0.0102859
```

```r
oopar <- par(mfrow=c(1,2))
xlim <- range(c(MC$res, boot$t[,1]))

#Histograma para permutacion boostrap
hist(MC$res[-length(MC$res)], main="Permutation bootstrap", xlab=expression(I[std]), xlim=xlim, density=15,
angle=45, ylim=c(0,260))
abline(v=MC$statistic, lty=2)

## Histograma para parametric boostrap
hist(boot$t, col=rgb(0.4,0.4,0.4), main="Parametric bootstrap", xlab=expression(I[CR]), xlim=xlim, ylim=c(0,
50))
hist(MC$res[-length(MC$res)], density=15, angle=45, add=TRUE)
abline(v=boot$t0, lty=2)
```




```r
# Plot the distribution (note that this is a density plot instead of a histogram)
plot(MC, main="", las=1)
```



Question 6: Write your own Geary C function, by completing the function below

```
#gearyC <- ((n-1)/sum(( "----")\^2)) * sum(wm * (" --- ")\^2) / (2 * sum(wm))

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


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
```