Informe laborario 5-Mayker Elizondo Ureña

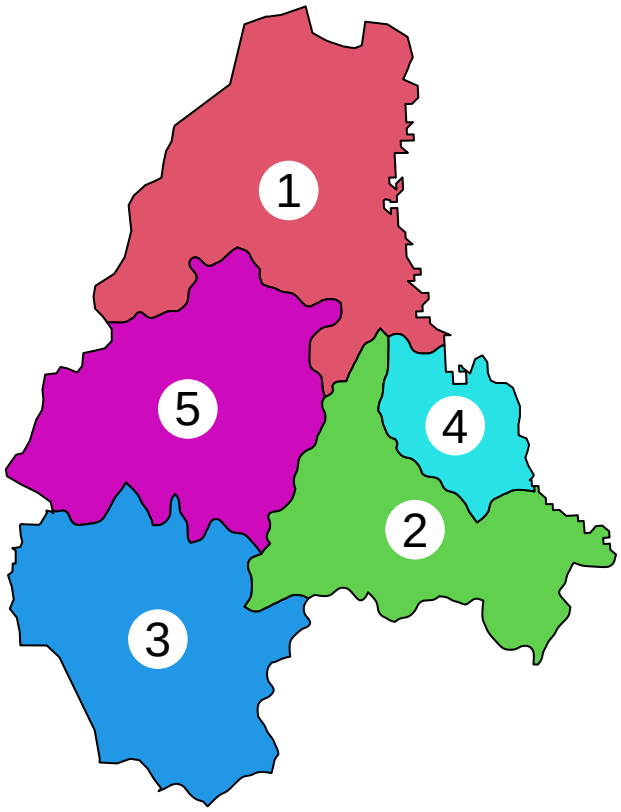
Intruducción

La autocorrelación espacial en estadistica es un fenomeno que no se puede dejar de lado, cuando se están construyendo modelos se debe valorar este fenomeno para la valoración de los supuestos o bien puede utlizarce para interpolaciones

Especificamente las medidas de autocorrelación espacial describen el grado en que dos observaciones (valores) en ubicaciones espaciales (ya sean puntos, áreas o celdas ráster) son similares entre sí

Para el laboratorio se utilizan datos espaciales devididos en 5 regiones y se busca determinar la correlación espacial de la variable value, que es una medida de estas regiones.

La distribución de los datos y las regiones se puede apreciar en el siguiente gráfico:



Polígonos adyacentes

Especificamente, se tiene como objetivo determinar las regiones que están "cerca" entre sí y como cuantificarlo.

Incialmente se hace el cálculo de las regiones que están cercas entre sí, el resultado se aprecia en la siguiente salida, en donde está la primer pregunta del laboratorio

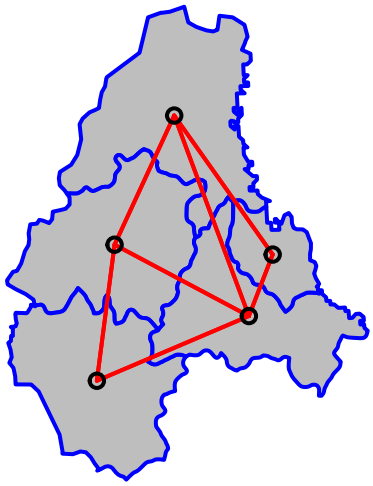
Pregunta 1 : Explique el significado de las primeras 5 líneas devueltas en el siguiente str (w)

```
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
```

Las primeras cinco lineas los que nos indican son los vecinos de cada región, por ejemplo, la región 1 tiene de vecinos al 2, 4, 5, la región 2 tiene de vecinos a 1,3,4,5 es decir todas las demas regiones

La interpratación anterior, se puede apreciar en el siguiente gráfico:



El siguiente paso en el laboratorio es calcular el I de moran

Pregunta 2 : ¿Cómo interpreta estos resultados (las pruebas de significancia) de los siguientes test de moran?

Primeramente, de la siguiente prueba se puede decir que el valor esperado del de I de moran es de -0.25, y el valor obtenido es de 0.1728896 y la variancia es de 0.03, con estos tres elementos se puede hacer la prueba estadística o bien o se puede analizar el p-value calculado que es de 0.009714 y con un nivel de significacia del 5% se rechaza la hipotesis de nula de no correlación en los datos

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

```

    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
```

Por otra parte, se hace la prueba pero mediante una simulación de montecarlo, lo cual ese hace en el siguiente comandos, de donde se puede interpretar que con un nivel de significancia del 5%, al comparar el P-value este es menor y se rezacha la hipotesis nula de no correlacion 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 = 99, p-value = 0.01
alternative hypothesis: greater
```

Por último, se prueba el siguinte código: **moran.mc(p$value, ww, nsim=999)**, el cual da un error, es porque el numero de simulaciones es muy grande, mayor que total de permutaciones posibles con 5 regiones

```
moran.mc(p$value, ww, nsim=999)
```

```
Error in moran.mc(p$value, ww, nsim = 999): nsim too large for this number of observations
```

Esto lleva a la siguinte pregunta: ***Pregunta 3 : ¿Cuál es el valor máximo que podemos usar para nsim?***

Para determinar la cantidad máxima de simulaciones, es importante tener en cuenta que para el ejercicio de montecarlo lo que se hace es seleccionar todas las cantidas posibles de permutaciones y con 5 regiones el número de permutaciones es de 120, en el siguiente código se puede apreciar la prueba con 120 y con 121

Prueba con 120

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

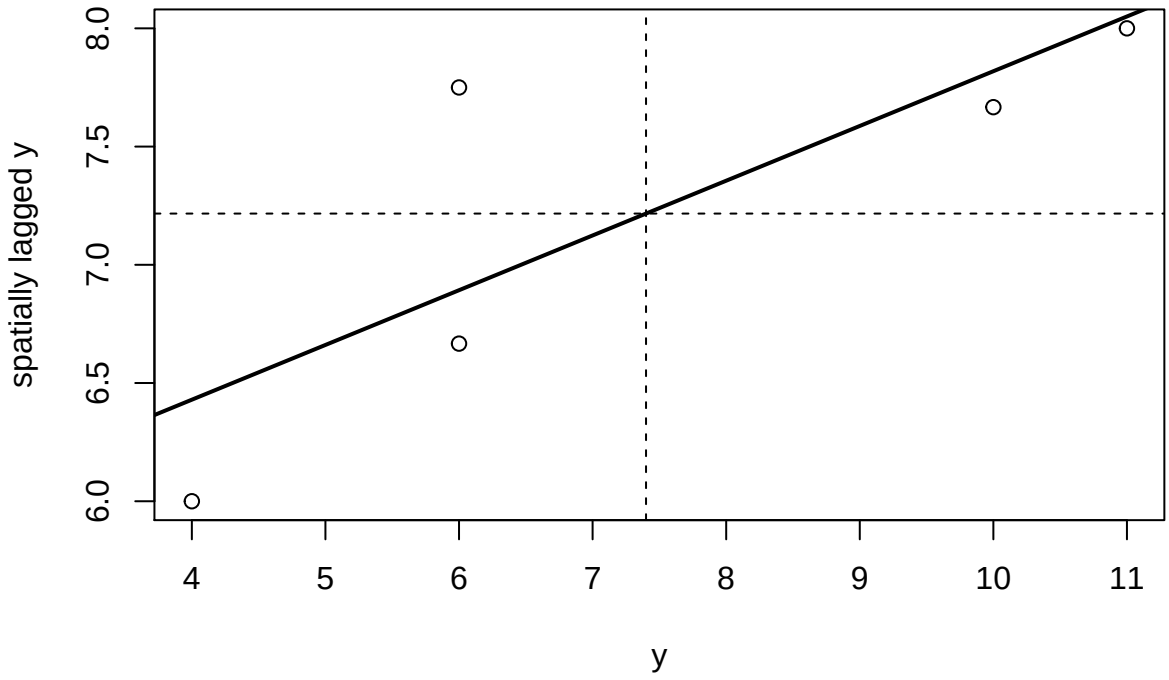
```
    Monte-Carlo simulation of Moran I

data:  p$value
weights: ww
number of simulations + 1: 121

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

Prueba con 121

```
moran.mc(p$value, ww, nsim=121)
```

```
Error in moran.mc(p$value, ww, nsim = 121): nsim too large for this number of observations
```

Con la I de moran, se puede hacer un gráfico de dispersión y precisamente en lo que se presenta en el siguiente gráfico:



Por último se plantean las siguientes preguntas en el laboratorio:

Pregunta 4 : Muestre cómo usar la función 'geary' para calcular la C de Geary

En el siguiene código se aprecia como se utiliza la función ***Geary*** a la cual se le suministra el valor de la variable a media, en este caso "value", luedo la matriz de pesos "WW", por último las cantidad de elementos elementos del ejercicio.

```
geary(p$value, ww, length(ww$neighbours), length(ww$neighbours)-1,
      Szero(ww))
```

```
$C
[1] 0.5357143

$K
[1] 1.432464
```

Pregunta 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

```
promedio=mean(y)
sd=sd(y)
v=rnorm(5,0,1)
moran.mc(v,ww,nsim = 120)
```

```
    Monte-Carlo simulation of Moran I
```

```
data:  v
weights: ww
number of simulations + 1: 121

statistic = -0.36056, observed rank = 33, p-value = 0.7273
alternative hypothesis: greater
```

```
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
```

Pregunta 6 : Escriba su propia función Geary C, completando la función a continuación

gearyC <- ((n-1)/sum(("—-")^2)) **sum(wm * (" — ")^2) / (2 * sum(wm))***

En las siguientes líneas se establecen los parametros utilizados en la formula

```
n <- length(p)
y <- p$value
ybar <- mean(y)
gg <- expand.grid(y, y)
yy_dif <- gg[,1] - gg[,2]
```

ya en el siguiente código se establece la formula final:

```
gearyC_COMPLE<- ((n-1)/sum( ((y-ybar)^2) ))*sum(wm * (gg[,1] - gg[,2])^2) / (2 * sum(wm))
gearyC_COMPLE
```

```
[1] 0.5357143
```

Es importante destacar que el valor es el mismo que el obtenido mediante la formula de: ***geary***