

Laboratorio 2: Reporte del artículo Introductory Point Pattern Analysis of Open Crime Data in London

Tópicos de Estadística Espacial Aplicada

Katherine Barquero

24/9/2020

Paso 1: Código y su comprensión

Se instalan los paquetes requeridos

Lectura del archivo

- Se utiliza el archivo enviado por la profesora
- Se muestra la estructura de los datos y se encuentra que las dimensiones de los datos difieren de las indicadas en el artículo que aparece con 79 832 observaciones y 12 variables

```
#class(data.London$Crime.type)
#Lectura del archivo
data <- read.csv("C:/Users/Katherine/OneDrive/UCR/Maestria/Geoespacial/Laboratorios/Lab2/london_street.csv", stringsAsFactors = TRUE)
str(data) #Ojo, en el enlace (http://r-video-tutorial.blogspot.com/2015/05/introductory-point-pattern-analysis-of.html) aparece 79 382 observaciones y 12 variables
```

```
## 'data.frame':    5000 obs. of  14 variables:
## $ X                : int  86142 35052 81151 35159 70976 29351 79002 39235 50390 75126 ...
## $ Crime.ID          : Factor w/ 3498 levels "", "0000f7649e0fd2cb2fcdd63062d54d0fd46ca54bc80442af218d43ecd511848e",...: 2206 1584 2096 1 1 1 1 605 3198 2366 ...
## $ Month             : Factor w/ 1 level "2014-06": 1 1 1 1 1 1 1 1 1 1 ...
## $ Reported.by       : Factor w/ 2 levels "City of London Police",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ Falls.within      : Factor w/ 2 levels "City of London Police",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ Longitude         : num  -0.195 -0.111 -0.253 -0.106 -0.142 ...
## $ Latitude          : num  51.4 51.4 51.5 51.4 51.4 ...
## $ Location          : Factor w/ 3500 levels "On or near A1201",...: 1584 3023 2531 1647 169 246 2624 2277 11 61 234 ...
## $ LSOA.code         : Factor w/ 2609 levels "E01000001", "E01000002",...: 2117 490 1981 497 1608 347 1900 681 990 1752 ...
## $ LSOA.name         : Factor w/ 2609 levels "Barking and Dagenham 001A",...: 2213 537 2039 543 1693 365 1961 678 1032 1835 ...
## $ Crime.type        : Factor w/ 14 levels "Anti-social behaviour",...: 11 11 7 1 1 1 1 13 14 14 ...
## $ Last.outcome.category: Factor w/ 20 levels "", "Court case unable to proceed",...: 13 6 6 1 1 1 1 6 18 6 ...
## $ Context           : logi  NA NA NA NA NA NA ...
## $ optional          : logi  TRUE TRUE TRUE TRUE TRUE TRUE TRUE ...
```

Se eliminan las coordenadas geográficas con valores NA

Se requiere contar con un set de datos que cuente con el tipo de crimen y coordenadas geográficas sin valores perdidos. Se observa que al eliminar los valores perdidos la dimensión original del set de datos no cambia.

```
data <- data[!is.na(data$Longitude)&!is.na(data$Latitude),]
dim(data) # Notar que la dimensión no cambia
```

```
## [1] 5000    14
```

Trabajar el archivo para realizar el análisis de un proceso puntual

Detección de duplicidades

Se requiere transformar el data frame a un objeto en formato Spatstat pero primero se deben eliminar las ubicaciones que están duplicadas. + Se transforman los datos originales a un Objeto espacial + Se eliminan las duplicidades en las ubicaciones. Se encuentran 585 duplicidades. + Debe encontrarse una forma de eliminarlas

```
coordinates(data)=~Longitude+Latitude #Transformación a un objeto espacial
zero <- zerodist(data) # Zerodiste elimina detecta si hay ubicaciones duplicadas
length(unique(zero[,1])) #Cantidad de ubicaciones duplicadas
```

```
## [1] 585
```

Eliminar crímenes de otras áreas

- Se carga primero los límites con la información de Natural Earth

```
download.file("http://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/ne_10m_admin_1_states_provinces.zip",destfile="ne_10m_admin_1_states_provinces.zip")
unzip("ne_10m_admin_1_states_provinces.zip",exdir="NaturalEarth")
border <- shapefile("NaturalEarth/ne_10m_admin_1_states_provinces.shp")
```

- Se extraen los bordes del área que interesa en el análisis: La región de Londres

```
GreaterLondon <- border[paste(border$region)=="Greater London",]
```

- Se sobreponen los datos de los crímenes a los bordes del área de Londres extraídos en el punto anterior y se eliminan los puntos que no pertenecen al área de análisis
- Se debe garantizar que ambos archivos (los datos de crimen y el objeto creado con los bordes de la región de análisis) tengan la misma proyección de los datos. Ambos poseen la proyección WGS84 por lo que se pueden sobreponer y garantizar que se eliminarán los puntos correctos.

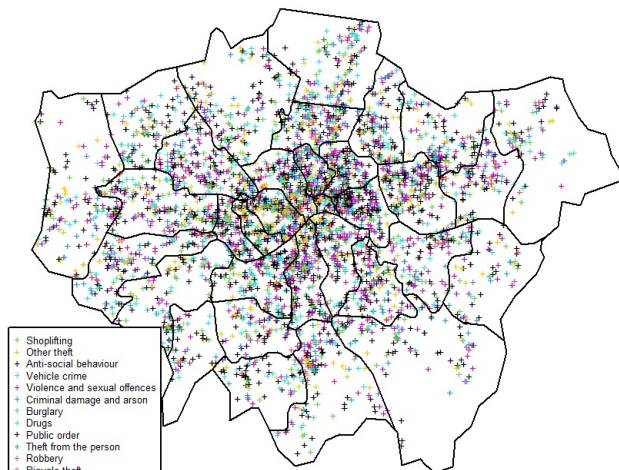
```
projection(data)=projection(border) # Se asigna la proyección de los bordes de Londres a los datos de Crimen
```

```
overlay <- over(data,GreaterLondon) # Se sobreponen los datos de crimen a la región de análisis
data$over <- overlay$adm1_code
```

```
data.London<- data[is.na(data$over)==F,]
# Se eliminan los puntos que no pertenecen al área de análisis
```

Graficar los datos sobrepuestos

```
plot(data.London,pch="+",cex=0.5,main="",col=data.London$Crime.type)
plot(GreaterLondon,add=T)
legend(x=-0.53,y=51.41,pch="+",col=unique(data.London$Crime.type),legend=unique(data.London$Crime.type),cex=0.4)
```



Análisis descriptivo de los datos

- Para el análisis que se realiza se requiere examinar la distribución espacial de los eventos y realizar inferencias sobre los patrones que se encuentren
- Se estiman los índices

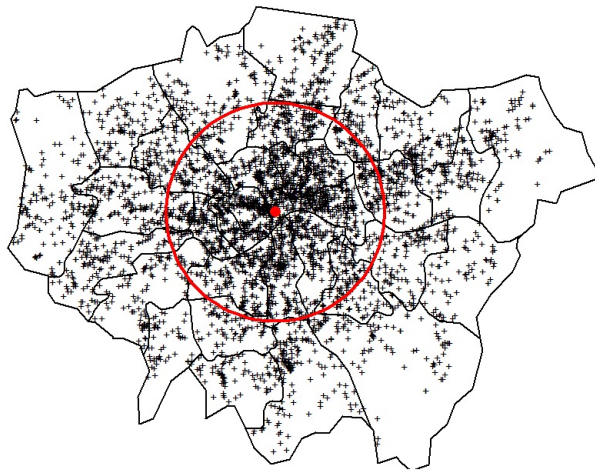
```
mean_centerX <- mean(data.London@coords[,1])
mean_centerY <- mean(data.London@coords[,2])

standard_deviationX <- sd(data.London@coords[,1])
standard_deviationY <- sd(data.London@coords[,2])

standard_distance <- sqrt(sum(((data.London@coords[,1]-mean_centerX)^2+(data.London@coords[,2]-mean_centerY)^2))/(nrow(data.London)))
```

- Se procede a graficar la dispersión de los datos con respecto a la media utilizando la distancia estándar
- Para esto se grafica el círculo de la media

```
#jpeg("PP_Circle.jpeg",2500,2000,res=300)
plot(data.London,pch="+",cex=0.5,main="")
plot(GreaterLondon,add=T)
points(mean_centerX,mean_centerY,col="red",pch=16)
draw.circle(mean_centerX,mean_centerY,radius=standard_distance,border="red",lwd=2)
```



```
#dev.off()
```

- Se grafica nuevamente pero utilizando la elipse de la media en lugar del círculo, esto permite incluir las desviaciones estándar de longitud y latitud mientras que en el círculo se promediaban y no se visualizaba de forma correcta las dimensiones

```
#jpeg("PP_Ellipse.jpeg",2500,2000,res=300)
plot(data.London,pch="+",cex=0.5,main="")
plot(GreaterLondon,add=T)
points(mean_centerX,mean_centerY,col="red",pch=16)
draw.ellipse(mean_centerX,mean_centerY,a=standard_deviationX,b=standard_deviationY,border="red",lwd=2)
```



```
#dev.off()
```

Eliminar los duplicados

- Se crean subsets por tipos de delitos para poder eliminar los duplicados.

- Solo se toman en cuenta los delitos por drogas
- Se obtiene un dataset con 1 375 eventos alrededor de Londres

```
Drugs <- data.London[data.London$Crime.type==unique(data.London$Crime.type)[3],]
Drugs <- remove.duplicates(Drugs)
dim(Drugs)
```

```
## [1] 1375 13
```

Se define la ventana o área donde se ubican las observaciones bajo análisis + Se requiere determinar cuántos eventos se tienen en una ventana (área) predeterminada. Se debe calcular la intensidad y densidad de los eventos + Lo primero es transformar el archivo en UTM

```
library(maptools)
#Transform GreaterLondon in UTM
GreaterLondonUTM <- spTransform(GreaterLondon,CRS("+init=epsg:32630"))
Drugs.UTM <- spTransform(Drugs,CRS("+init=epsg:32630"))

#Transforming the SpatialPolygons object into an owin object for spatstat, using a function in maptools

window <- as.owin(GreaterLondonUTM)
```

Estimar la densidad e intensidad de los eventos

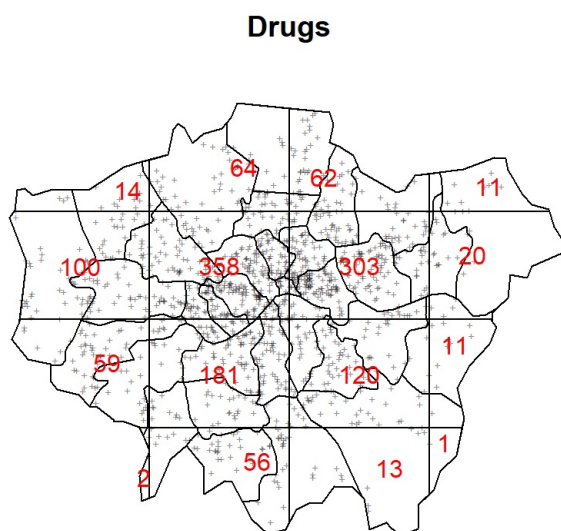
- La intensidad se define como el número de eventos promedio por unidad de área

```
Drugs.ppp <- ppp(x=Drugs.UTM@coords[,1],y=Drugs.UTM@coords[,2],window=window)
Drugs.ppp$n/sum(sapply(slot(GreaterLondonUTM, "polygons"), slot, "area"))
```

```
## [1] 8.566451e-07
```

- Se grafica la intensidad de los eventos para determinar si el proceso bajo análisis es uniforme o homogéneo
- Para determinarlo en este caso se utiliza el recuento en cuadrantes (quadrat counting), que divide el número de eventos en cada rectángulo

```
#jpeg("PP_QuadratCounting.jpeg",2500,2000,res=300)
plot(Drugs.ppp,pch="+",cex=0.5,main="Drugs")
plot(quadratcount(Drugs.ppp, nx = 4, ny = 4),add=T,col="red")
```



```
#dev.off()
```

- Para dar mas precisión al análisis se extrae la información por división administrativa de la región de análisis esto produce más certeza y precisión de la cantidad de crimes o delitos en cada una de ellas

```
Local.Intensity <- data.frame(Borough=factor(),Number=numeric())
for(i in unique(GreaterLondonUTM$name)){
  sub.pol <- GreaterLondonUTM[GreaterLondonUTM$name==i,]

  sub.ppp <- ppp(x=Drugs.ppp$x,y=Drugs.ppp$y>window=as.owin(sub.pol))
  Local.Intensity <- rbind(Local.Intensity,data.frame(Borough=factor(i,levels=GreaterLondonUTM$name),Number=sub.ppp$
n))
}
```

```
## Warning: 1320 points were rejected as lying outside the specified window
```

```
## Warning: 1351 points were rejected as lying outside the specified window
```

```
## Warning: 1301 points were rejected as lying outside the specified window
```

```
## Warning: 1339 points were rejected as lying outside the specified window
```

```
## Warning: 1329 points were rejected as lying outside the specified window
```

```
## Warning: 1325 points were rejected as lying outside the specified window
```

```
## Warning: 1338 points were rejected as lying outside the specified window
```

```
## Warning: 1350 points were rejected as lying outside the specified window
```

```
## Warning: 1365 points were rejected as lying outside the specified window
```

```
## Warning: 1298 points were rejected as lying outside the specified window
```

```
## Warning: 1337 points were rejected as lying outside the specified window
```

```
## Warning: 1324 points were rejected as lying outside the specified window
```

```
## Warning: 1326 points were rejected as lying outside the specified window
```

```
## Warning: 1339 points were rejected as lying outside the specified window
```

```
## Warning: 1362 points were rejected as lying outside the specified window
```

```
## Warning: 1357 points were rejected as lying outside the specified window
```

```
## Warning: 1347 points were rejected as lying outside the specified window
```

```
## Warning: 1330 points were rejected as lying outside the specified window
```

```
## Warning: 1314 points were rejected as lying outside the specified window
```

```
## Warning: 1345 points were rejected as lying outside the specified window
```

```
## Warning: 1322 points were rejected as lying outside the specified window
```

```
## Warning: 1334 points were rejected as lying outside the specified window
```

```
## Warning: 1313 points were rejected as lying outside the specified window
```

```
## Warning: 1331 points were rejected as lying outside the specified window
```

```
## Warning: 1337 points were rejected as lying outside the specified window
```

```
## Warning: 1333 points were rejected as lying outside the specified window
```

```
## Warning: 1331 points were rejected as lying outside the specified window
```

```
## Warning: 1356 points were rejected as lying outside the specified window
```

```
## Warning: 1307 points were rejected as lying outside the specified window
```

```
## Warning: 1348 points were rejected as lying outside the specified window
```

```
## Warning: 1311 points were rejected as lying outside the specified window
```

```
## Warning: 1344 points were rejected as lying outside the specified window
```

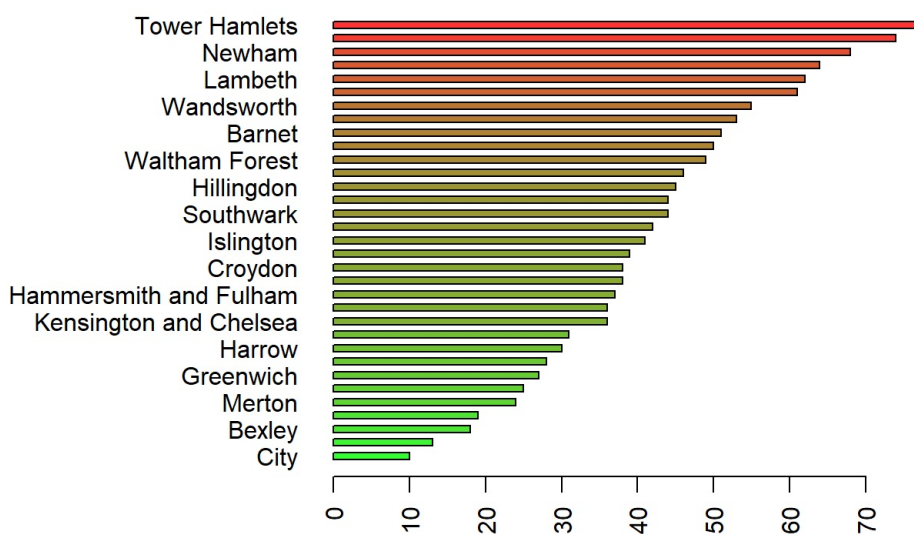
```
## Warning: 1336 points were rejected as lying outside the specified window
```

```
colorScale <- color.scale(Local.Intensity[order(Local.Intensity[,2]),2],color.spec="rgb",extremes=c("green","red"),alpha=0.8)
```

```
#jpeg("PP_BoroughCounting.jpeg",2000,2000,res=300)
```

```
par(mar=c(5,13,4,2))
```

```
barplot(Local.Intensity[order(Local.Intensity[,2]),2],names.arg=Local.Intensity[order(Local.Intensity[,2]),1],horiz=T,las=2,space=1,col=colorScale)
```



```
#dev.off()
```

- Otra forma para analizar la intensidad de la distribución espacial es a partir de la técnica de suavizamiento de Kernel que calcula la intensidad de forma continua en toda el área de estudio
- Se debe definir el ancho de banda de la estimación de la densidad que determina el área de influencia
- Los gráficos muestran distintos anchos de banda y en función de su tamaño se pueden generar distintas interpretaciones

```
#jpeg("Kernel_Density.jpeg",2500,2000,res=300)
```

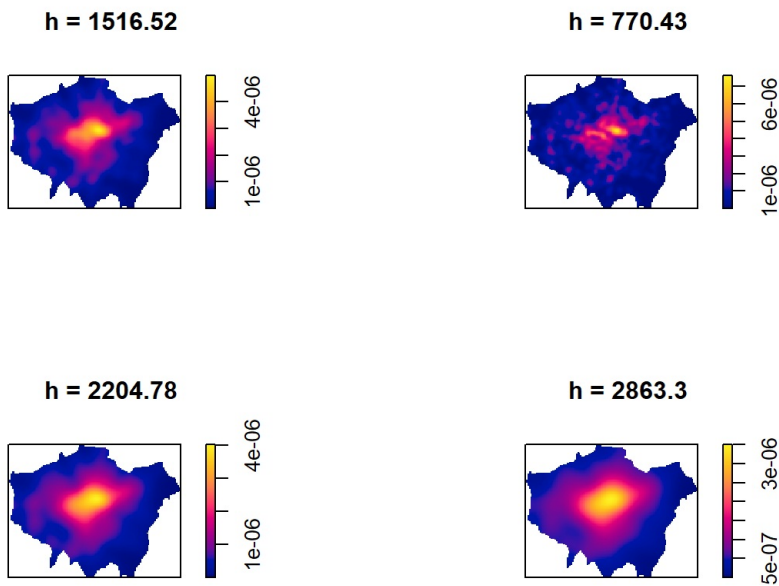
```
par(mfrow=c(2,2))
```

```
plot(density.ppp(Drugs.ppp, sigma = bw.diggle(Drugs.ppp),edge=T),main=paste("h =",round(bw.diggle(Drugs.ppp),2)))
```

```
plot(density.ppp(Drugs.ppp, sigma = bw.ppl(Drugs.ppp),edge=T),main=paste("h =",round(bw.ppl(Drugs.ppp),2)))
```

```
plot(density.ppp(Drugs.ppp, sigma = bw.scott(Drugs.ppp)[2],edge=T),main=paste("h =",round(bw.scott(Drugs.ppp)[2],2)))
```

```
plot(density.ppp(Drugs.ppp, sigma = bw.scott(Drugs.ppp)[1],edge=T),main=paste("h =",round(bw.scott(Drugs.ppp)[1],2)))
```

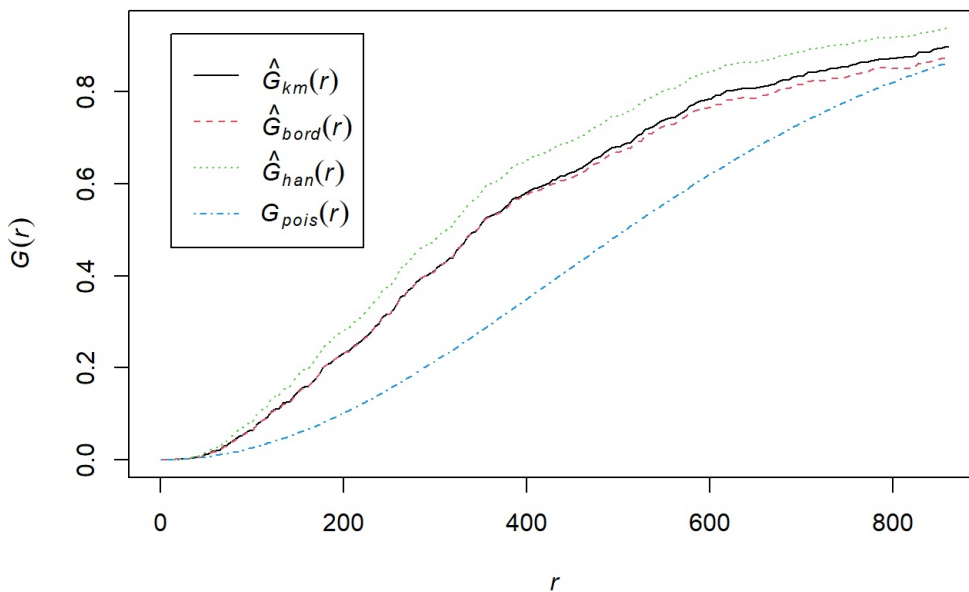


```
#dev.off()
```

- Se debe analizar si un patrón de datos es aleatorio (cada punto es independiente entre si y de cualquier otro factor). La aleatoriedad espacial completa implica que los eventos del proceso puntual tienen la misma probabilidad de ocurrir en todas las regiones de la ventana de estudio.
- Para determinar cuál es la distribución que tienen los datos se utiliza la función Gest del paquete Stat, la línea azul muestra el patrón para puntos espaciales aleatorios completos.

```
#jpeg("GFunction.jpeg",2500,2000,res=300)
plot(Gest(Drugs.ppp),main="Drug Related Crimes")
```

Drug Related Crimes



```
?Gest
```

```
## starting httpd help server ... done
```

```
#dev.off()
```

Paso 2: Informe

El estudio de caso pretendía responder a las siguientes preguntas de investigación:

¿Cuál es la distribución espacial que presentan los eventos por delitos de drogas en Londres? ¿Siguen un patrón aleatorio o de otro tipo?

Para dar respuesta a estas interrogantes se utilizaron los datos sobre delitos de los meses de mayo y junio de 2014 para toda Gran Bretaña, y se realizó un análisis de procesos puntuales el cual es un proceso estocástico donde se observan los eventos específicos de análisis que ocurren en una región o área específica para determinar si estos presentan un patrón en su comportamiento aleatorio, agrupado o de otro tipo.

Para esto se desarrollaron los siguientes pasos:

1. Definir el software y los paquetes requeridos para el análisis

- Se selecciona el paquete R
- Se identifican las librerías a utilizar: sp, raster, spatstat, maptools and plotrix.

2. Preparar y transformar los datos para poder manipularlos a través de las librerías seleccionadas

- Verificar que el set de datos contengan toda la información completa relacionada con la ubicación espacial (coordenadas geográficas)
- Definir el área de análisis
- Eliminar ubicaciones duplicadas: sobreponer el archivo a un objeto geográfico que cuente con la misma proyección de datos y eliminar los puntos que no se ubican en el área de análisis
- Graficar los objetos sobrepuestos

3. Estimar la densidad e intensidad de los eventos que ocurren en la región de análisis

- Se estiman las estadísticas descriptivas: media-centro y las desviaciones estándar de las coordenadas geográficas y la distancia estándar
- Se analiza la dispersión de los datos alrededor de su centro medio: Con círculo y elipse. En este último se grafican las desviaciones estándar de longitud y latitud mientras que en el círculo se promedian y por tanto no se visualiza de forma correcta las dimensiones.
- Se estima la intensidad y densidad de los eventos a través de: método de cuadrantes, extracción de los datos por división administrativa, o a partir de la Estimación de Kernel a partir del ancho de banda.

4. Identificar el tipo de distribución espacial que presentan los eventos de análisis + Se estima la función G y se determina si pueden haber agrupaciones en los datos o si el patrón de los datos es aleatorio.