Laboratorio 2

Mariana Cubero

24/9/2020

El ejercicio de análisis de procesos puntuales que se toma como referencia para este laboratorio tiene como objetivos responder a las siguientes preguntas de investigación:

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

Para dar respuesta a estas interrogantes se planteanse realiza un análisis de procesos puntuales. Este análisis es un proceso estocástico donde se observan los eventos específicos de interés al 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.

Los paquetes de y herramientas utilizadas son las siguientes:

```
# Definir directorio de trabajo
setwd("C:/Users/maric/OneDrive/maestria/Geoestadistica/SP1649-
II20/Laboratorio 2")
library(spatstat)
```

```
## Loading required package: spatstat.data
```

```
## Loading required package: nlme
```

```
## Loading required package: rpart
```

```
## 
## spatstat 1.64-1       (nickname: 'Help you I can, yes!') 
## For an introduction to spatstat, type 'beginner'
```

```
## 
## Note: spatstat version 1.64-1 is out of date by more than 4 months; we
recommend upgrading to the latest version.
```

```
library(sp)
library(rgdal)
```

```
## rgdal: version: 1.5-12, (SVN revision 1018)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.0.4, released 2020/01/28
## Path to GDAL shared files: C:/Users/maric/Documents/R/win-
library/4.0/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ_VERSION: 631]
## Path to PROJ shared files: C:/Users/maric/Documents/R/win-
library/4.0/rgdal/proj
## Linking to sp version:1.4-2
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading
rgdal.

library(raster)

##
## Attaching package: 'raster'

## The following objects are masked from 'package:spatstat':
##
##     area, rotate, shift

## The following object is masked from 'package:nlme':
##
##     getData

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:raster':
##
##     intersect, select, union

## The following object is masked from 'package:nlme':
##
##     collapse

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(maptools)

## Checking rgeos availability: TRUE

# Cargar datos
data <- read.csv("london_street.csv", stringsAsFactors =TRUE)
str(data)

## '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 1161 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 ...
```

Preparación de los datos para el análisis

Como es común, los datos que se quieren estudiar no vienen en con la estructura y formato requerido para el análisis. Por esto se consideran las siguientes transformaciones y ajustes para realizar el análisis.

Convertir las tablas de cols a coordenadas

En el caso de estos datos es importante destacar que no son un objeto espacial formal, si no que se construye a partir de dos vectores de coordenas. Hay que tener cuidado y definir una proyección adecuada para evitar errores principalmente porque se van a hacer cálculos.

```
coordinates(data)=~Longitude+Latitude
```

Para locaciones duplicadas

Estos duplicados se deben eliminar porque el paquete spatstats no “aguanta” puntos sobrepuestos. Si bien por la naturaleza de los datos es posible que un crime ocurra en el mismo punto se deben tomar en cuenta problemas de registro entre otros. Por eso se usa la función “zerodist” que identifica puntos que tenga distancia cero entre ellos es decir que están encima.

```
zero <- zerodist(data)
length(unique(zero[,1]))
```

```
## [1] 585

#Note que el N es diferente al artículo
```

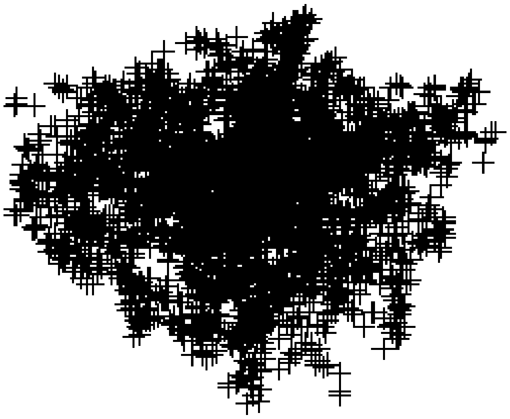
Cargar shape de fondo

Este shape se usa para jalar el sistema de referencia "CRS" para conectarlo con las coordenadas de los crímenes.

```
#download.file("http://www.naturalearthdata.com/http//www.naturalearthdata.co
m/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")
```

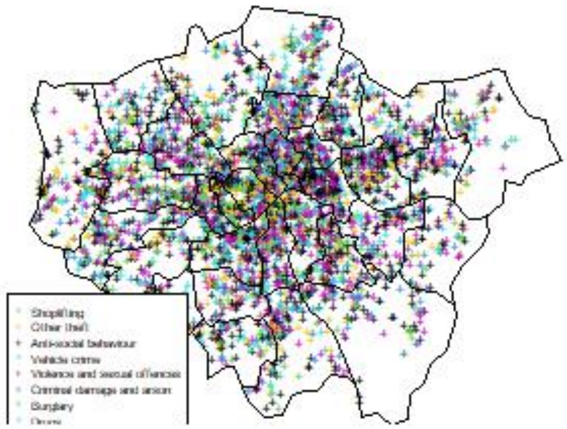
Subset de una parte de londres

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



projection(data)=projection(border)
overlay <- over(data,GreaterLondon)

data$over <- overlay$adm1_code

data.London <- data[is.na(data$over)==F,]
plot(data.London)
```



Graficar

```
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.Lo
ndon$Crime.type),cex=0.4)
```



Análisis descriptivo

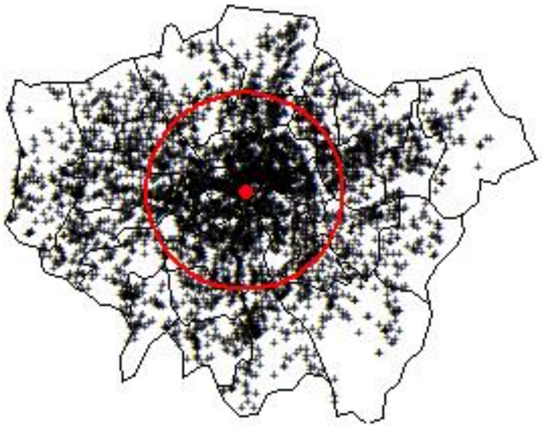
Calcular los indices

```
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)))
```

Graficar el círculo de la media

```
plot(data.London,pch="+",cex=0.5,main="")
plot(GreaterLondon,add=T)
points(mean_centerX,mean_centerY,col="red",pch=16)
```

```
plotrix::draw.circle(mean_centerX,mean_centerY,radius=standard_distance,borde
r="red",lwd=2)
```



Hacer un subset de los crimenes por tipo, solo se va a tomar en cuenta el tipo

crímenes por drogas.

```
library(dplyr)
Drugs <- data.London %>% subset(Crime.type == "Drugs") %>%
remove.duplicates()
head(Drugs@data)

##         X                                                         Crime.ID
## 23  35407 40e35d83381b449cece9878ec54273ce3fcb06d8690fdff18dbd7ba9523b79b4
## 67  80840 3b79e81c4fb1de93f3182844f9754230d75864c7a427f13f47195da9d7c575e6
## 109 35996 a6326b5cca4eca5ffa98b138682b9207fe7ac05cedc8cd133c0a491efd683ea0
## 149 29118 ec1d9db584d2236570b205ca3ea8233a655d259703112e3b594bf4b546e6f6ae
## 150 78099 12f634095b986b8488f184df8c09d4408845907df56d2ff9670b806e67d8ceb9
## 156 87225 87ad170c9964c43007709cd53500cc569201835de6ec31d1c76c5922f87a627e
##       Month                 Reported.by                  Falls.within
## 23  2014-06 Metropolitan Police Service Metropolitan Police Service
## 67  2014-06 Metropolitan Police Service Metropolitan Police Service
## 109 2014-06 Metropolitan Police Service Metropolitan Police Service
## 149 2014-06 Metropolitan Police Service Metropolitan Police Service
## 150 2014-06 Metropolitan Police Service Metropolitan Police Service
## 156 2014-06 Metropolitan Police Service Metropolitan Police Service
##                         Location LSOA.code          LSOA.name Crime.type
## 23      On or near Parking Area E01001118       Croydon 020E      Drugs
## 67  On or near Ranelagh Gardens E01003692     Redbridge 035C      Drugs
## 109        On or near Fell Road E01001043       Croydon 027B      Drugs
## 149   On or near Freelands Road E01000825       Bromley 008G      Drugs
## 150   On or near Basin Approach E01003482        Newham 033B      Drugs
## 156    On or near Shopping Area E01004234 Tower Hamlets 003E      Drugs
##                             Last.outcome.category Context optional
over
## 23      Offender given a drugs possession warning      NA     TRUE GBR-
2065
## 67      Offender given a drugs possession warning      NA     TRUE GBR-
2074
## 109                                Offender fined      NA     TRUE GBR-
2065
## 149 Investigation complete; no suspect identified      NA     TRUE GBR-
2063
## 150             Offender given community sentence      NA     TRUE GBR-
2770
## 156                      Offender given a caution      NA     TRUE GBR-
2765
```

#Definir la ventana donde están las observaciones bajo estudio

```
#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 <- spatstat::as.owin(GreaterLondonUTM)
```

La intensidad se define como el número de eventos promedio por unidad o cuadrante en la ventana. Para calcular la intensidad es así:

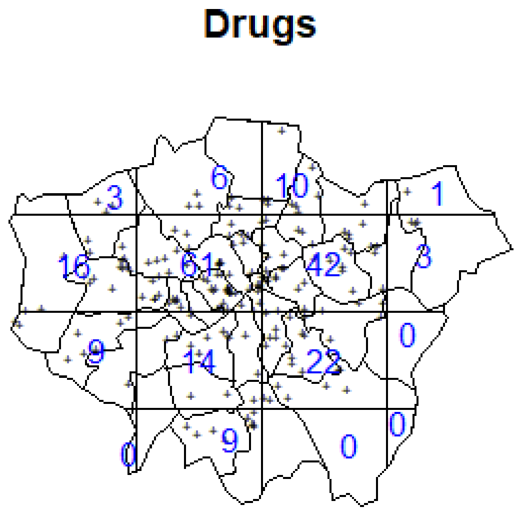
```
Drugs.ppp <-
spatstat::ppp(x=Drugs.UTM@coords[,1],y=Drugs.UTM@coords[,2],window=window)
Drugs.ppp$n/sum(sapply(slot(GreaterLondonUTM, "polygons"), slot, "area"))

## [1] 1.221109e-07

Drugs.ppp$n/sum(sapply(slot(GreaterLondonUTM, "polygons"), slot, "area"))

## [1] 1.221109e-07
```

Análisis PP

Intesidad y densidad

El problema es que este gráfico no funciona bien porque sólo dibuja un punto.

```
plot(Drugs.ppp,pch="+",cex=0.5,main="Drugs")
plot(spatstat::quadratcount(Drugs.ppp, nx = 4, ny = 4),add=T,col="blue")
```

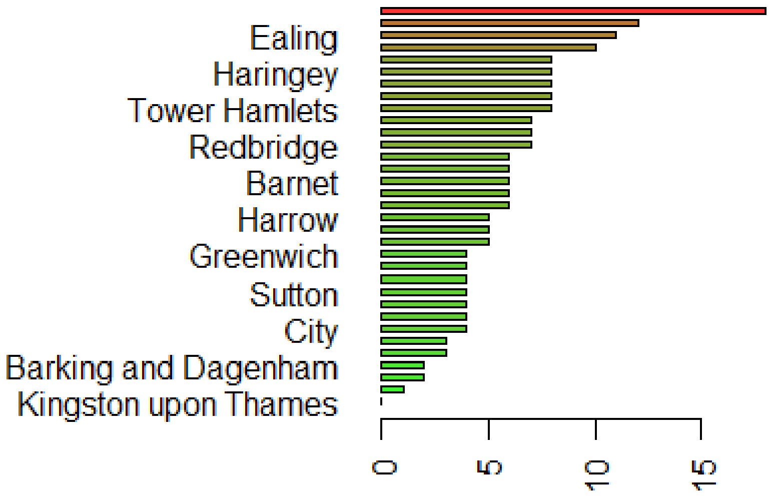


Se extrae la información por división administrativa de la región de análisis esto produce más certeza y precisión de la cantidad de crímenes en cada una. Se usa un ciclo para este calculo. Esta función lo que hace es calcular la intensidad de crímenes (concentración de puntos) en cada cuadrante para posteriormente visualizarlo de una manera más funcional para los datos ya que el mapa no hace mucho snetido para interpretación. O la indentificación de patrones.

```
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$nam
e),Number=sub.ppp$n))
}
```

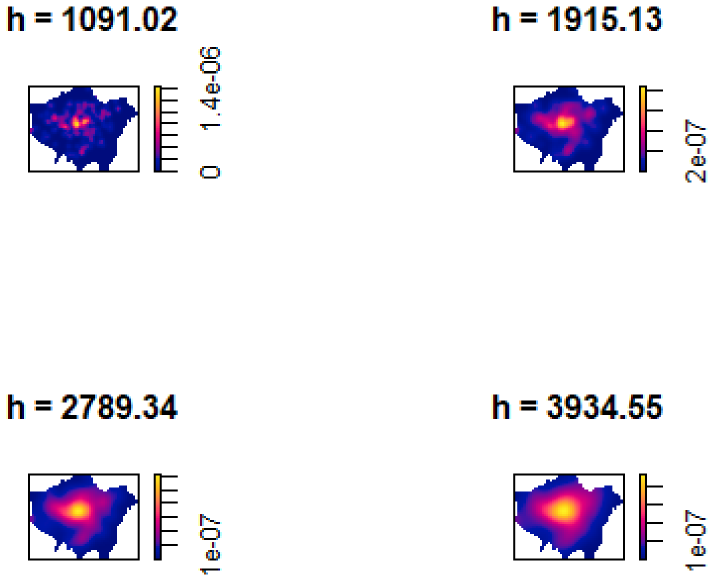
```
## Warning: 191 points were rejected as lying outside the specified window
## Warning: 194 points were rejected as lying outside the specified window

## Warning: 178 points were rejected as lying outside the specified window

## Warning: 193 points were rejected as lying outside the specified window

## Warning: 190 points were rejected as lying outside the specified window

## Warning: 185 points were rejected as lying outside the specified window

## Warning: 190 points were rejected as lying outside the specified window

## Warning: 193 points were rejected as lying outside the specified window

## Warning: 192 points were rejected as lying outside the specified window

## Warning: 188 points were rejected as lying outside the specified window

## Warning: 192 points were rejected as lying outside the specified window

## Warning: 190 points were rejected as lying outside the specified window

## Warning: 188 points were rejected as lying outside the specified window

## Warning: 189 points were rejected as lying outside the specified window

## Warning: 192 points were rejected as lying outside the specified window

## Warning: 195 points were rejected as lying outside the specified window

## Warning: 192 points were rejected as lying outside the specified window

## Warning: 191 points were rejected as lying outside the specified window

## Warning: 184 points were rejected as lying outside the specified window

## Warning: 191 points were rejected as lying outside the specified window

## Warning: 189 points were rejected as lying outside the specified window

## Warning: 192 points were rejected as lying outside the specified window

## Warning: 192 points were rejected as lying outside the specified window

## Warning: 186 points were rejected as lying outside the specified window

## Warning: 188 points were rejected as lying outside the specified window

## Warning: 190 points were rejected as lying outside the specified window

## Warning: 188 points were rejected as lying outside the specified window

## Warning: 196 points were rejected as lying outside the specified window

## Warning: 189 points were rejected as lying outside the specified window
## Warning: 192 points were rejected as lying outside the specified window

## Warning: 188 points were rejected as lying outside the specified window

## Warning: 194 points were rejected as lying outside the specified window

## Warning: 190 points were rejected as lying outside the specified window
```

```
colorScale <-
plotrix::color.scale(Local.Intensity[order(Local.Intensity[,2]),2],color.spec
="rgb",extremes=c("green","red"),alpha=0.8)
par(mar=c(5,13,4,2))
barplot(Local.Intensity[order(Local.Intensity[,2]),2],names.arg=Local.Intensi
ty[order(Local.Intensity[,2]),1],horiz=T,las=2,space=1,col=colorScale)
```



Técnica de suavisamiento de kernel

```
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)))
```

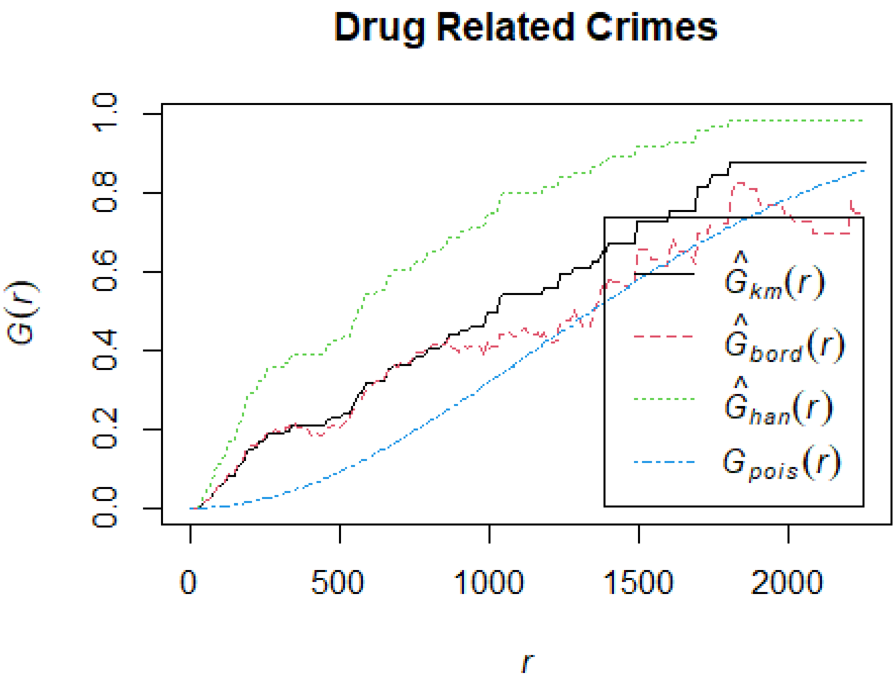


El objetivo del análisis era determinar si los eventos (crímenes) eran aleatorios o seguían alguna distribución. La aleatoriedad espacial completa implica que los eventos del proceso puntual tienen la misma probabilidad de ocurrir en todas las regiones de la ventana de estudio.

Otra forma de analizar los resultados es la siguiente, usando la función Gest. Donde la línea azul muestra el patrón para puntos espaciales aleatorios completos.

Los datos no se parecen a ninguna de las distribuciones de comparación.

```
plot(Gest(Drugs.ppp),main="Drug Related Crimes")
```


Drug Related Crimes
G(r)
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
r
$\hat{G}_{km}(r)$
$\hat{G}_{bord}(r)$
$\hat{G}_{han}(r)$
$G_{pois}(r)$