Lab 2-B12337

Metodos de investigación y preguntas de investigación

El primer autor inicia con unos delitos de Londres, en una primera instancia se enfrenta al problema de observar observaciones repetidas para un mismo delito, en donde menciona que no tiene suficiente información para saber cual es la forma adecuada de tratarlo, en el sentido que se puede tratar que un delito está en dos categorías o un delito fue cometido por dos personas distintas y otras posibles opciones. Otro de los puntos que desarrolla el autor y parte del supuesto que es, es más interesante investigar los delitos de Gran Londres, el centro de Londres, para esto utiliza un shapefile que tiene la delimitación del área en donde él quiere trabajar.

Posterior y delimitar la información del Gran Londres, inicia con un análisis descriptivo de la información, en este punto inicialmente saca un promedio 2D, que es básicamente un promedio de la longitud y un promedio de la latitud, además cálcula un indicador de desviación estándar también en 2D. En este punto es muy interesante en el sentido que con el promedio se obtiene el centro en donde se ubican los delitos y con la desviación estándar un radio que da una idea de la dispersión y que especialmente en el mapa genera mucho valor.

Es importante destacar que en el punto de graficar la desviación estandar, se tienen dos opciones, primera fue graficar este indicador con la fórmula general, la segunda hacer grafico con la desviación de la latitud y longitud por aparte, en el primer caso se obtiene un circulo y en el segundo un eclipse.

El siguiente paso que dio el autor fue centrarse puramente en los delitos de drogas y además de eliminar los duplicados esto porque asumió que se deben a un mismo delito.

En este siguiente paso trabajó con el Spatstat,

En este parte de la investigación lo que busca son patrones de puntos, que los define con puntos en una ventana o área determinada. Para este paso se apoyó en la funciónowin para encontrar esas ventanas.

Un patrón de puntos se define como una serie de eventos en un área o ventana de observación determinada. Por tanto, es extremadamente importante definir con precisión esta ventana.

Posterior de haber definido esas ventanas menciona que lo que corresponde es definir una medida de intensidad en cada una de las ventanas, para esto comenta que la medida adecuada es cantidad de delitos pos metros cuadrados, para esto hizo una transformación de los datos y posteriormente hizo el cálculo de la métrica.

En un siguiente paso el autor indica que lo interesante consiste generar la métrica de cantidad de delitos para los “condados” en esa linea lo que hace es un hacer un ciclo para sacar la cantidad y la delimitación de cada “condado”. A partir de este condado se obtiene un grafico de barras en donde se observa que brent es donde hay una mayor cantidad de delitos por metro cuadrado, seguidamente de Lambeth.

Posterior a obtener esta grafico de barras, mediante un función de kernel también genera la intensidad, en este punto menciona que para la función de kernel se debe definir el ancho de banda y que no hay alguna regla, si h es muy alto se pueden perder elementos importantes y si es muy pequeño se puede hacer muy ruidosa.

Aleatoriedad espacial completa

Después de haber definido los puntos, el autor menciona que es importante definir si hay aleatoriedad o no, define no alteridad como la ubicación de puntos con una densidad mayor que la promedio para esto se apoya en la función G del paquete Spatstat. con esta función lo que se busca es encontrar la distribución de los datos y con ello ver si responde o no un proceso aleatorio.

En general concluye que no es un proceso aleatorio y que hay patrones que dan evidencia de que en ciertos lugares se comenten más delitos. En resumen el autor se plantea investigar patrones espaciales de los delitos de Gran Londres, inicialmente con todos los datos sacó medidas descriptivas para todos los delitos, luego se enfocó solo en los delitos de drogas, en este punto da un siguiente paso y saca la cantidad de delitos por metro cuadrado, es decir la densidad, para esto busca el mundo ideal y es hacerlo por "condado", en donde obtiene el top con mayor densidad, luego con una función de kernel genera gráficos de tipo de calor y se aprecia claramente una patrón espacial en los delitos de drogas y por lo último o confirma con función G que arroja que como resultado que los datos no se distribuyen de manera aleatorio, sino que responden a un patrón espacial.

Código Librerias

```
suppressPackageStartupMessages(library(sp))
suppressPackageStartupMessages(library(spatstat))
suppressPackageStartupMessages(library(raster))
suppressPackageStartupMessages(library(maptools))
suppressPackageStartupMessages(library(plotrix))
suppressPackageStartupMessages(library(rgeos))
suppressPackageStartupMessages(library(readr))
```

```
data <- read_csv("C:/Users/User/Documents/Mayker/Maestria Profesional/Estadistica Espacial/Lab 2/london_
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   Crime.ID = col_character(),
##   Month = col_character(),
##   Reported.by = col_character(),
##   Falls.within = col_character(),
##   Longitude = col_double(),
##   Latitude = col_double(),
##   Location = col_character(),
##   LSOA.code = col_character(),
##   LSOA.name = col_character(),
##   Crime.type = col_character(),
##   Last.outcome.category = col_character(),
##   Context = col_logical(),
##   optional = col_logical()
## )
```

```
data <- data[!is.na(data$Longitude)&!is.na(data$Latitude),]
str(data)
```

```
## Registered S3 method overwritten by 'cli':
##   method     from
##   print.boxx spatstat
```

```
## tibble [5,000 x 14] (S3: tbl_df/tbl/data.frame)
##  $ X1                  : num [1:5000] 86142 35052 81151 35159 70976 ...
##  $ Crime.ID            : chr [1:5000] "a0feda8c2ab111cd313b875520387d493b14f82e546afd687e725737d667a
##  $ Month               : chr [1:5000] "2014-06" "2014-06" "2014-06" "2014-06" ...
##  $ Reported.by         : chr [1:5000] "Metropolitan Police Service" "Metropolitan Police Service" "M
##  $ Falls.within        : chr [1:5000] "Metropolitan Police Service" "Metropolitan Police Service" "M
##  $ Longitude           : num [1:5000] -0.195 -0.111 -0.253 -0.106 -0.142 ...
##  $ Latitude            : num [1:5000] 51.4 51.4 51.5 51.4 51.4 ...
```

```
##  $ Location             : chr [1:5000] "On or near High Street" "On or near Supermarket" "On or near
##  $ LSOA.code            : chr [1:5000] "E01004140" "E01001005" "E01003857" "E01001013" ...
##  $ LSOA.name            : chr [1:5000] "Sutton 012B" "Croydon 019A" "Richmond upon Thames 003F" "Cro
##  $ Crime.type           : chr [1:5000] "Shoplifting" "Shoplifting" "Other theft" "Anti-social behavi
##  $ Last.outcome.category: chr [1:5000] "Offender given penalty notice" "Investigation complete; no su
##  $ Context              : logi [1:5000] NA NA NA NA NA NA ...
##  $ optional             : logi [1:5000] TRUE TRUE TRUE TRUE TRUE TRUE ...
```

```
coordinates(data)=~Longitude+Latitude
zero <- zerodist(data)
length(unique(zero[,1]))
```

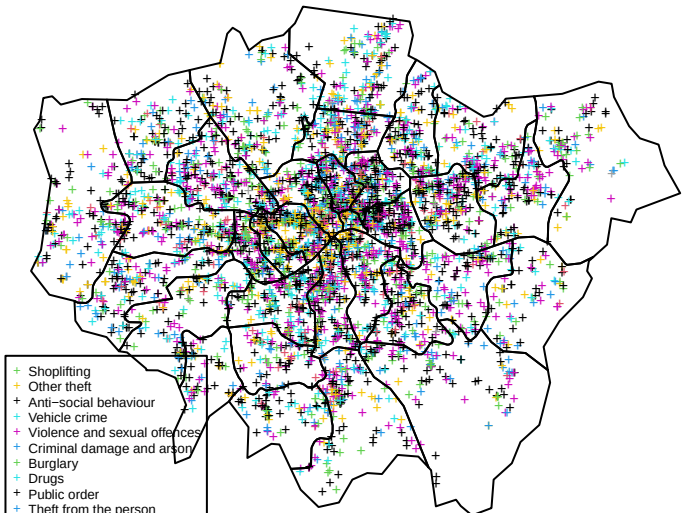
```
## [1] 585
```

En el siguiente código lo que se hace descargar es descargar un shp que tiene la delimitación de Gran Londres y así trabajar solo con está area, después de haber descargado el arhivo, los delitos que quedan con NA posterior hacer el cruce son eliminados son elimados ya que son los que no están en en Gran Londres. Por último se hace el grafico del área utilizada con la cantidad de delitos.

```
# download.file("http://www.naturalearthdata.com/http//www.naturalearthdata.com/download/10m/cultural/n
unzip("ne_10m_admin_1_states_provinces.zip",exdir="NaturalEarth")
border <- shapefile("NaturalEarth/ne_10m_admin_1_states_provinces.shp")
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),]
#jpeg("PP_plot.jpg",2500,2000,res=300)
summary(data.London$Crime.type)
```

```
##    Length     Class      Mode
##      4955 character character
```

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




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

En este punto está sacando el promedio para la altitud y la latitud y luuego saca la desviación estándar.

```
#Summary statistics for point patterns
#The coordinates of the mean center are simply the mean value of X and Y
#therefore we can use the function mean() to determine their value
mean_centerX <- mean(data.London@coords[,1])
mean_centerY <- mean(data.London@coords[,2])
```

Saca la desvición estándar para cada cordenada

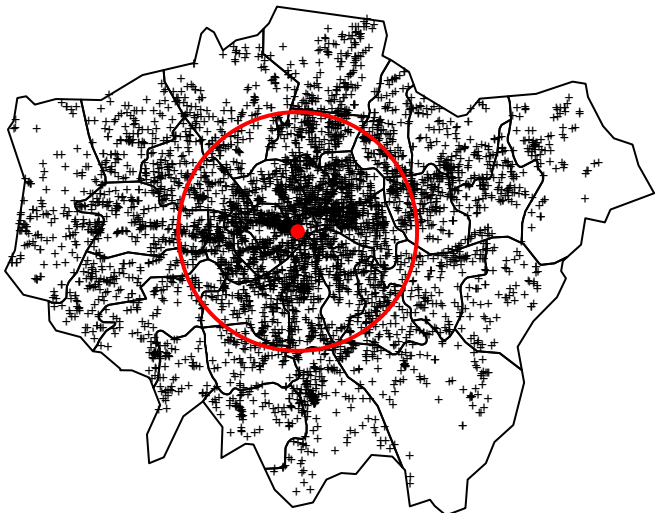
```
#Similarly we can use the function sd() to determine the standard deviation of X and Y
standard_deviationX <- sd(data.London@coords[,1])
standard_deviationY <- sd(data.London@coords[,2])
```

En este punto genera ya una desviación estandar unificada.

```
#This is the formula to compute the standard distance
standard_distance <- sqrt(sum(((data.London@coords[,1]-mean_centerX)^2+(data.London@coords[,2]-mean_cen
```

En el siguiente código hace un gráfico de la área y los puntos de los delitos, pero además le agrega el promedio de los puntos y la desviación estandar general cálculada en el punto anterior.

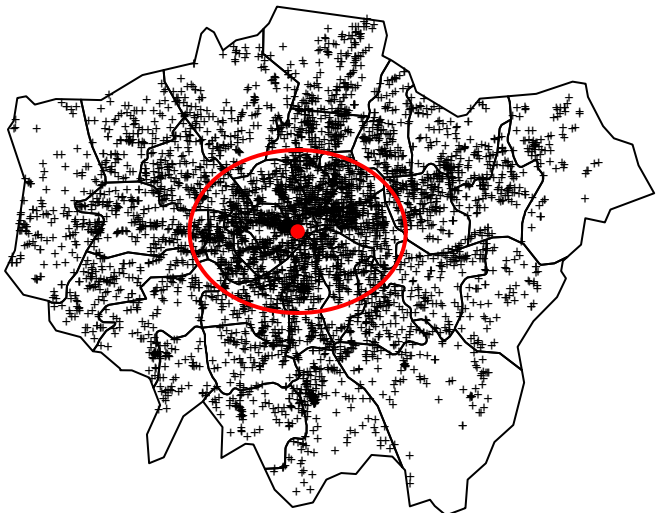
```
#jpeg("PP_Circle.jpeg",2500,2000,res=300)
plot(data.London,pch="+",cex=0.5,main="")
plot(GreaterLondon,add=T)
points(mean_centerX,mean_centerY,col="red",pch=16)
draw.circle(mean_centerX,mean_centerY,radius=standard_distance,border="red",lwd=2)
```



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

El autor menciona que en el punto anterior ambas desviaciones reciben el mismo peso, al poner cada desviación por aparte ya gráfico un elipse

```
#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()
```

El siguiente peso fue eliminar los valores repetidos y además trabajar solo con los delitos de drogas, eso se hace en el siguiente código

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

Para sacar un indicar de densidad de delitos por metro cuadrado hace la siguiente transformación .

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

Estos indicaores de densidad se cálculan es un espacio determinado y para eso se utiliza la funcion owin, para calcular la ventanas.

```
#Transforming the SpatialPolygons object into an owin object for spatstat, using a function in maptools
window <- as.owin(GreaterLondonUTM)
```

en el siguiente código incorpora esas ventanas en el los datos

```
#Now we can extract one crime and
Drugs.ppp <- ppp(x=Drugs.UTM@coords[,1],y=Drugs.UTM@coords[,2],window=window)
```

Se cálcula el relación de delitos por metro cuadrado.

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

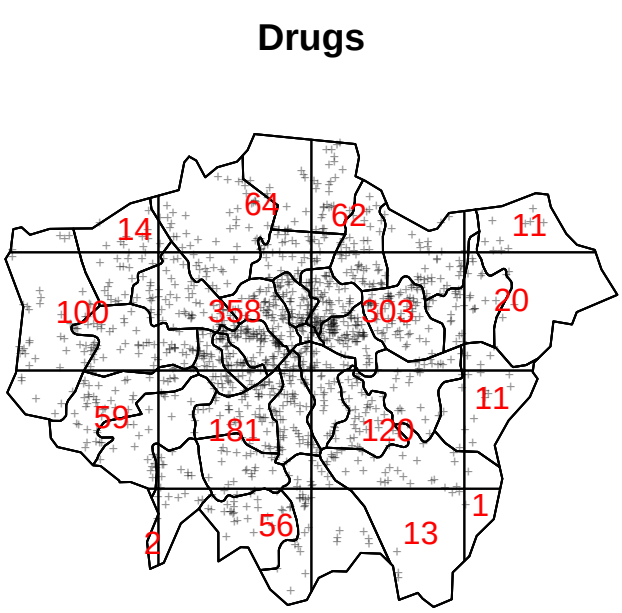
```
## [1] 8.566451e-07
```

```
#Alternative approach
summary(Drugs.ppp)$intensity
```

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

El siguiente paso fue gráficar la indicador de delitos por metros cuadrado y eso se hace en el siguiente código

```
#Quadrat counting Intensity
#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()
```

El autor menciona que las areas tiene sentido que sean cálcudas según los “condados” y precidamente eso cálcula en siguiente código, mediante un ciclo hace el cálculo de delitos por “condados”

```
#Intensity by Borough
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),Nu
}
```

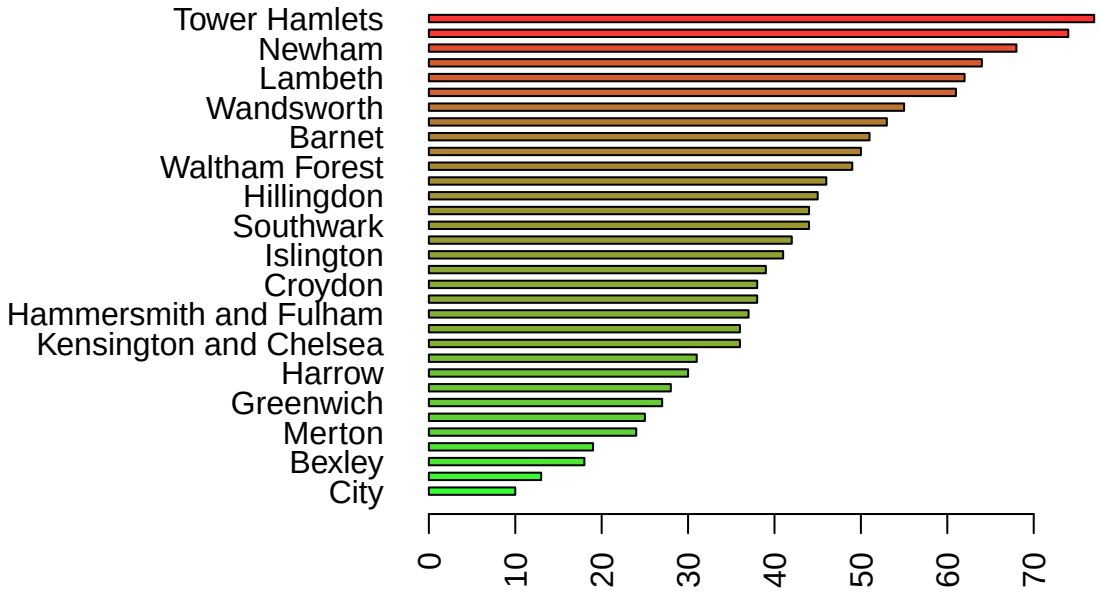
```
## Warning: 1320 points were rejected as lying outside the specified window
```

```
## 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
```

graficamente el tip de condados con mayores delitos por metro cuadrado se hace en el siguiente código y se observa en el siguiente gráfico.

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

#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[,
```




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

Otra de las técnicas que usó el autor fue mediante una funcion de kernel calcular la desindad y precisamente eso hace en las siguientes líneas

```
#Kernel Density (from: Baddeley, A. 2008. Analysing spatial point patterns in R)
#Optimal values of bandwidth
bw.diggle(Drugs.ppp)
```

```
##    sigma
## 1516.522
```

```
bw.ppl(Drugs.ppp)
```

```
##    sigma
## 770.4251
```

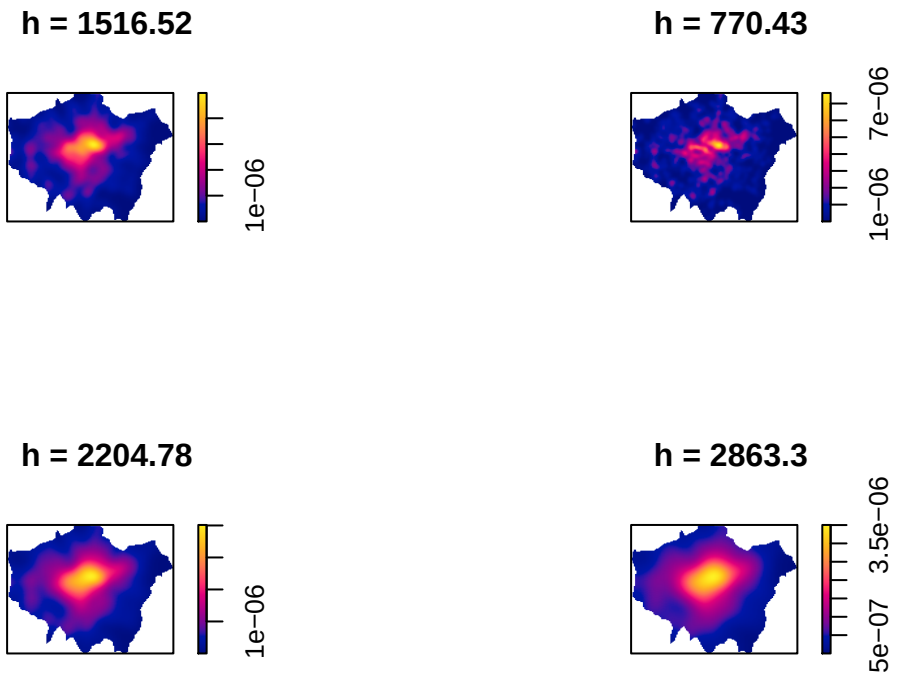
```
bw.scott(Drugs.ppp)
```

```
##  sigma.x  sigma.y
## 2863.297 2204.782
```

Gráficamente los resultados de la función de kernel se observa en las siguientes líneas y se aprecia que si hay

un patron espacial en los delitos cometidos

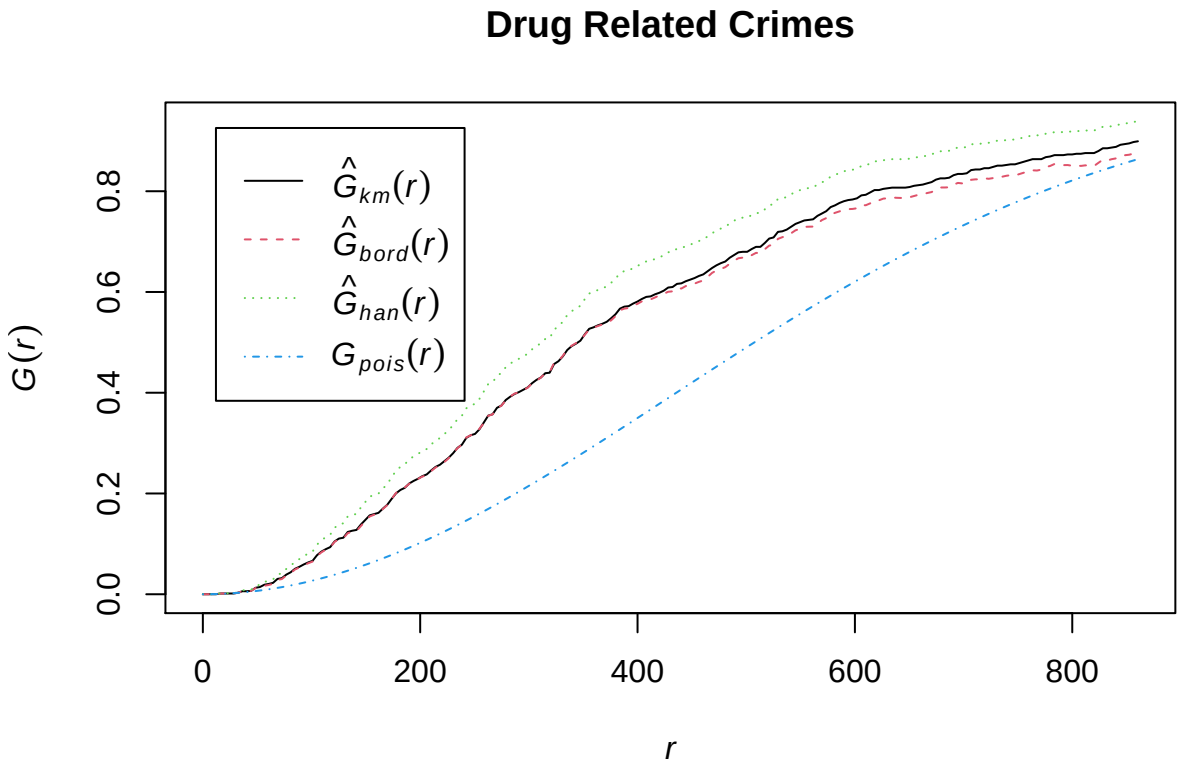
```
#Plotting
#jpeg("Kernel_Density.jpeg",2500,2000,res=300)
par(mfrow=c(2,2))
plot(density.ppp(Drugs.ppp, sigma = bw.diggle(Drugs.ppp),edge=T),main=paste("h =",round(bw.diggle(Drugs
plot(density.ppp(Drugs.ppp, sigma = bw.ppl(Drugs.ppp),edge=T),main=paste("h =",round(bw.ppl(Drugs.ppp),
plot(density.ppp(Drugs.ppp, sigma = bw.scott(Drugs.ppp)[2],edge=T),main=paste("h =",round(bw.scott(Drug
plot(density.ppp(Drugs.ppp, sigma = bw.scott(Drugs.ppp)[1],edge=T),main=paste("h =",round(bw.scott(Drug
```




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

Por último, da un paso mas el autor y mediante la función G, calcula si los delitos tienen un patron aleatroio o no y los resultados se aprecian en el siguiente gráfico, el cual es que no es un patrón aleatorio y todo lo contrario hay un patron espacial.

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


Drug Related Crimes
$\hat{G}_{km}(r)$
$\hat{G}_{bord}(r)$
$\hat{G}_{han}(r)$
$G_{pois}(r)$
G(r)
0.0
0.2
0.4
0.6
0.8
0
200
400
600
800
r

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