*Spatial Statistics Lab 4*

**Onyedikachi J Okeke**

2024-03-11

Q1 Kernel Density   
A screenshot of a computer

Description automatically generated

Figure 1: Kernel Density for crime\_hitandrun

A screenshot of a map

Description automatically generated

Figure 2: Kernel Density for crime\_disorder

A screenshot of a map

Description automatically generated

Figure 3: Kernel Density for crime\_burglary

# Q2 Average Nearest Neighbor

A screenshot of a computer

Description automatically generated

Average Nearest Neighbor

=====================

Parameters

Input Feature Class crime\_burglary

Distance Method EUCLIDEAN\_DISTANCE

Generate Report GENERATE\_REPORT

Area

Nearest Neighbor Index 0.832969

z-score -2.928647

p-value 0.003404

Expected Mean Distance 0.691273

Observed Mean Distance 0.575809

Report File C:\Spatial Statistics Labwork\NearestNeighbor\_Result\_7792\_4732\_.html

=====================

The pattern described in the Average Nearest Neighbor Summary indicates a clustering of the crime\_burglary data points. The Nearest Neighbor Ratio (NNI) of 0.832969 is less than 1, which suggests that the burglary incidents are closer to each other than would be expected if they were randomly distributed in the study area.

This clustering pattern is further supported by the z-score of -2.928647, which is a statistically significant negative value falling well outside the range of what would be expected under a random distribution (as depicted by the normal distribution curve where the tails represent the critical values for significance levels). Since the z-score is beyond the critical value for a 0.01 significance level (usually a z-score less than -2.58 or greater than +2.58), this strongly rejects the null hypothesis of random distribution.

The p-value of 0.003404 confirms the significance of this result, indicating a high level of confidence (less than 0.5% probability of error) that the observed clustering pattern is not due to random chance.

In summary, the pattern is one where burglary incidents are more concentrated or clustered in space rather than being spread out or randomly located within the study area. Such information is crucial for strategic planning in crime prevention and resource allocation.

Q3: ***Multi-Distance Spatial Cluster Analysis (Ripleys K Function)***

A screenshot of a computer

Description automatically generated

Figure 4: Multi-Distance Spatial Cluster Analysis (Ripleys K Function)

y = -0.410135 + 1.159842 x  
R2 = 0.99163

A graph with a red line

Description automatically generated

The graph provided represents a Multi-Distance Spatial Cluster Analysis for crime burglary data, visualized through Ripley's K function, which assesses spatial patterns at varying distances. The plot reveals a near-perfect linear correlation between the observed and expected values, indicated by an R-squared value of 0.99163. This high degree of correlation suggests that the observed spatial pattern of crime burglaries is almost entirely explained by the model used in the analysis. The regression equation \( y = -0.410135 + 1.159842x \) indicates that as the observed value increases, the expected value also increases at a rate slightly higher than the observed, reflecting a clustering trend. Despite the negative intercept, which may not have a practical interpretation, the slope greater than one points towards a tendency of burglaries to cluster rather than occurring randomly. However, the analysis should be approached with caution, as overfitting could be a concern given the high R-squared value, and further statistical tests would be needed to establish the significance of the observed clustering.

### O Load the library

Package ‘**rgda’** and ‘**maptools**’ was removed from the CRAN repository. Formerly available versions can be obtained from the archive. Archived on 2023-10-16 at the request of the maintainer. So, I used ‘sf’ or ‘terra’ instead since my R is 4.2.2.

library(sp)  
library(spatstat)  
library(sf)  
library(spatstat.geom)

Library

## Loading required package: spatstat.data

## Loading required package: spatstat.geom

## spatstat.geom 3.2-8

## Loading required package: spatstat.random

## spatstat.random 3.2-2

## Loading required package: spatstat.explore

## Loading required package: nlme

## spatstat.explore 3.2-5

## Loading required package: spatstat.model

## Loading required package: rpart

## spatstat.model 3.2-8

## Loading required package: spatstat.linnet

## spatstat.linnet 3.1-3

##   
## spatstat 3.0-7   
## For an introduction to spatstat, type 'beginner'

## Linking to GEOS 3.11.2, GDAL 3.7.2, PROJ 9.3.0; sf\_use\_s2() is TRUE

### Loading city\_limits\_km and StLouisCrime2014

#using the sp function to load the city\_limits\_km.shp  
S\_sf <- st\_read("C:/Spatial Statistics Labwork/Lab4Data/city\_limits\_km.shp")  
W <- as.owin(S\_sf)  
plot(W, main = "owin")  
  
#load the StLouisCrime2014  
xy <- read.table("C:\\Spatial Statistics Labwork\\Lab4Data\\StLouisCrime2014.txt", header=TRUE, sep="\t")  
#first 10 row  
head(xy, n = 10)  
attach(xy)

##### Result

## Reading layer `city\_limits\_km' from data source   
## `C:\Spatial Statistics Labwork\Lab4Data\city\_limits\_km.shp'   
## using driver `ESRI Shapefile'

## Warning in CPL\_read\_ogr(dsn, layer, query, as.character(options), quiet, : GDAL  
## Error 1: PROJ: proj\_identify: C:\Program  
## Files\PostgreSQL\14\share\contrib\postgis-3.2\proj\proj.db contains  
## DATABASE.LAYOUT.VERSION.MINOR = 0 whereas a number >= 2 is expected. It comes  
## from another PROJ installation.

## Simple feature collection with 1 feature and 3 fields  
## Geometry type: POLYGON  
## Dimension: XY  
## Bounding box: xmin: 265.5428 ymin: 299.8415 xmax: 278.0294 ymax: 321.897  
## Projected CRS: NAD\_1983\_StatePlane\_Missouri\_East\_FIPS\_2401

## FID FID\_ X Y CRIME Month  
## 1 0 1 266.7461 306.3005 BURGLARY 2013-10  
## 2 1 2 267.1740 305.7309 BURGLARY 2014-03  
## 3 2 3 267.4133 305.7418 BURGLARY 2013-09  
## 4 3 4 267.7307 306.7532 BURGLARY 2014-08  
## 5 4 5 267.7706 307.0418 BURGLARY 2014-03  
## 6 5 6 268.0429 313.4292 BURGLARY 2013-09  
## 7 6 7 268.3970 313.1811 BURGLARY 2014-06  
## 8 7 8 268.6476 313.5435 BURGLARY 2014-05  
## 9 8 9 268.9404 301.9671 BURGLARY 2014-06  
## 10 9 10 268.9906 304.5185 BURGLARY 2014-07



Figure 5: Imported Polygon

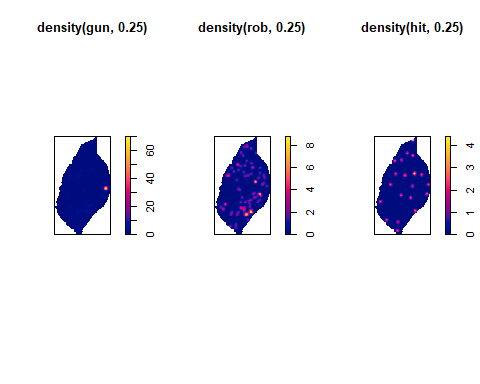
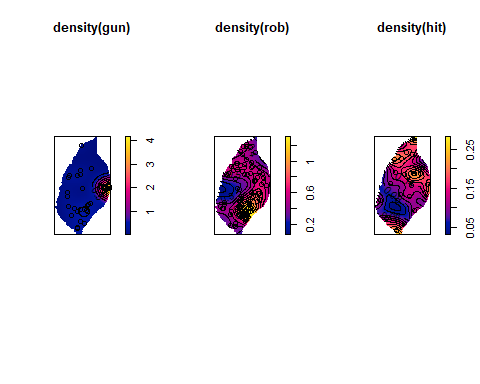
## converting to a spatstat point pattern object, with the different crime types as an identifying mark.

pp <- ppp(X, Y, window=W, marks=CRIME)  
  
plot(pp)  
  
#to work with each crime as a distinct dataset  
gun <- pp[CRIME=="DISORDERLY"]  
rob <- pp[CRIME=="BURGLARY"]  
hit <- pp[CRIME=="HITANDRUN"]  
  
par(mfrow = c(1, 3))  
#make density map for gun  
plot(density(gun))  
contour(density(gun), add=T)  
plot(gun, add=T)  
#make density map for rob  
plot(density(rob))  
contour(density(rob), add=T)  
plot(rob, add=T)  
#make density map for hit  
plot(density(hit))  
contour(density(hit), add=T)  
plot(hit, add=T)  
  
par(mfrow = c(1, 3))  
plot(density(gun, 0.25))  
plot(density(rob, 0.25))  
plot(density(hit, 0.25))

#### Result

## Warning: data contain duplicated points  
A map of a mountain

Description automatically generated

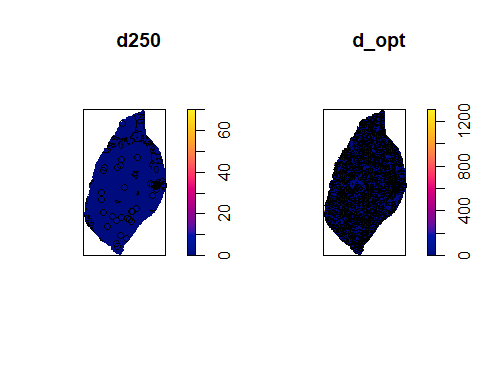


## Using the second parameter in the density function (bandwidth)

d250 <- density(gun, 0.25)  
  
par(mfrow = c(1, 2))  
  
plot(d250)  
contour(d250, add=T)  
plot(gun, add=T)  
  
#To R optimal bandwidth  
r <- bw.diggle(gun)  
r  
  
#apply the optimal bandwidth   
d\_opt <- density(gun, r)  
plot(d\_opt)  
contour(d\_opt, add=T)  
plot(gun, add=T)

#### Result

## sigma   
## 0.003054447



## Q4. Provide a commentary discussing the most suitable bandwidth choice for this analysis visualization method.

The provided plots display kernel density estimates for spatial data using different bandwidths. The first set of plots likely shows default bandwidth density estimates for varying incident types, while the second and third sets depict the effects of bandwidth choice on the density visualization for 'gun' data specifically.

The fixed bandwidth of 0.25, as seen in d250, smooths the data considerably, potentially obscuring finer details. Conversely, the optimal bandwidth (d\_opt), derived from bw.diggle(gun), produces a high-resolution density estimate that may be too fine-grained, possibly highlighting noise as patterns.

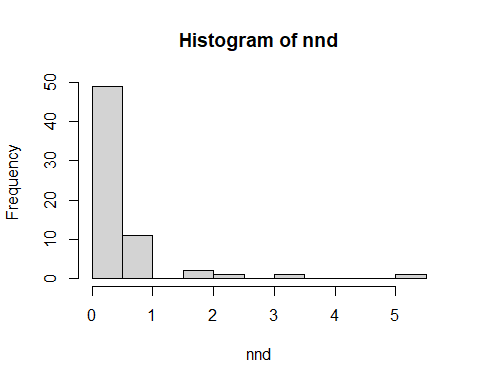
The optimal bandwidth, with sigma at 0.003054447, suggests a detailed but potentially overfitted pattern.   
Hence the suggested suitable bandwidth based on the data will be **0.02763.**

nnd <- nndist.ppp(gun)  
hist(nnd)  
summary(nnd)  
#compare the mean value to that expected for an IRP/CSR pattern of the same intensity  
mnnd <- mean(nnd)  
exp\_nnd <- 0.5 / sqrt(gun$n / area.owin(W))  
  
print (mnnd / exp\_nnd)

#### Result

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.1526 0.4584 0.4936 5.4488

## [1] 0.5757526



## Q5 Is this pattern clustered or evenly spaced?

The result of **mnnd / exp\_nnd** is 0.5757526, which is less than 1. This suggests that the observed point pattern has a higher density (i.e., points are closer together) than would be expected under CSR. A value less than 1 indicates clustering, as the actual nearest neighbor distances are shorter on average than expected if points were randomly distributed. In the context of selecting a suitable bandwidth for analysis or visualization, this clustering effect should be considered.

#spatstat are quadratcount() and quadrat.test():  
par(mfrow=c(1,1))  
q <- quadratcount(hit, 4, 8)  
plot(q)  
plot(hit, add=T)  
quadrat.test(hit, 4, 8)  
quadrat.test(hit, 4, 8, alternative="clustered")  
quadrat.test(hit, 4, 8, alternative="regular")

#### Result

## Warning: Some expected counts are small; chi^2 approximation may be inaccurate

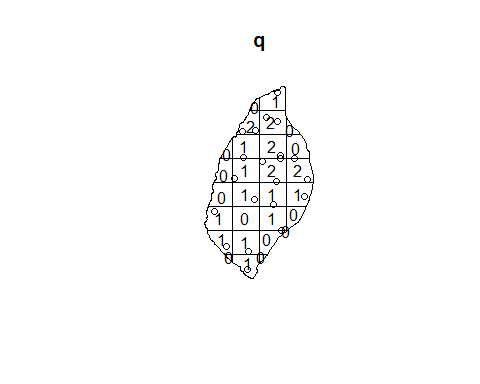
##   
## Chi-squared test of CSR using quadrat counts  
##   
## data: hit  
## X2 = 11.309, df = 27, p-value = 0.006984  
## alternative hypothesis: two.sided  
##   
## Quadrats: 28 tiles (irregular windows)

## Warning: Some expected counts are small; chi^2 approximation may be inaccurate

##   
## Chi-squared test of CSR using quadrat counts  
##   
## data: hit  
## X2 = 11.309, df = 27, p-value = 0.9965  
## alternative hypothesis: clustered  
##   
## Quadrats: 28 tiles (irregular windows)

## Warning: Some expected counts are small; chi^2 approximation may be inaccurate

##   
## Chi-squared test of CSR using quadrat counts  
##   
## data: hit  
## X2 = 11.309, df = 27, p-value = 0.003492  
## alternative hypothesis: regular  
##   
## Quadrats: 28 tiles (irregular windows)



Q6. Is it a clustered, regular, or random pattern?

Based on the chi-squared test results provided with the image, the spatial point pattern for 'hit' demonstrates a significant deviation from complete spatial randomness. The test indicates a p-value of 0.003492, which is well below the standard alpha level of 0.05, suggesting a **regular pattern**.

## The distance-based functions: G, F, K (and its relative L) and the more recent pair correlation function.

g\_gun <- Gest(gun)  
f\_gun <- Fest(gun)  
k\_gun <- Kest(gun)  
l\_gun <- Lest(gun)  
pcf\_gun <- pcf(gun)  
  
par(mfrow = c(1, 1))  
plot(g\_gun)  
plot(f\_gun)  
plot(k\_gun)  
plot(l\_gun)  
plot(pcf\_gun)  
  
#to calculate the function for a set of simulated realizations of IRP/CSR in the same study area.  
g\_gun\_env <- envelope(gun, Gest, nsim=99, nrank=1)  
  
plot(g\_gun\_env)

#### Result

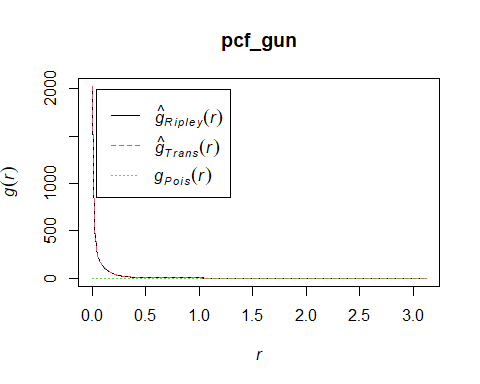
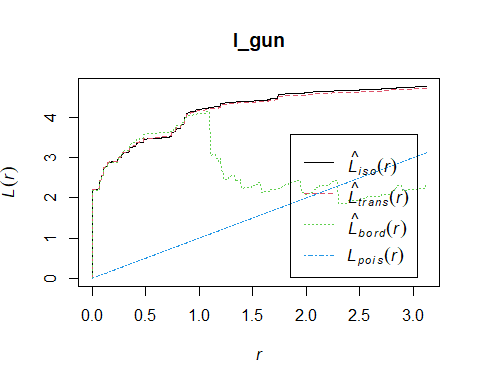
## Generating 99 simulations of CSR ...  
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,  
## 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,  
## 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,  
## 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,  
## 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,   
## 99.  
##   
## Done

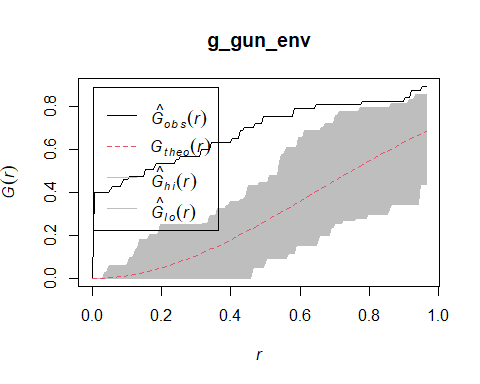
A graph with numbers and lines

Description automatically generated

A graph of an object

Description automatically generated



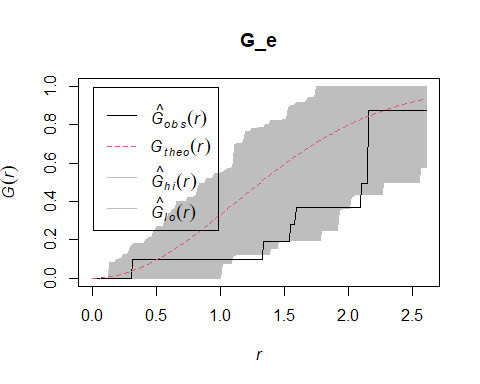


Q7. What does the plot show us?   
   
The plots represent spatial analyses of 'gun' data, comparing empirical spatial statistics to theoretical expectations under Complete Spatial Randomness (CSR). The `Gest` function plot shows values rising above `Gpois(r)`, indicating clustering at shorter distances. In the `Kest` function plot, values significantly above `Kpois(r)` suggest clustering at various scales. The `Lest` plot, where the empirical line is above the theoretical line `Lpois(r)`, also indicates clustering. The `pcf` plot exhibits a peak above `Gpois(r)` at short distances, reinforcing evidence of clustering. These suggest a non-random pattern, likely clustering, as supported by envelope simulations generated by `envelope(gun, Gest, nsim=99)`, confirming non-randomness at a 99 simulation level. The plots and simulations together suggest that 'gun' events are not randomly distributed but show signs of clustering.

G\_e <- envelope(hit, Gest, nsim=99, nrank=5)  
plot(G\_e)

#### Result

## Generating 99 simulations of CSR ...  
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,  
## 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,  
## 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,  
## 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,  
## 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98,   
## 99.  
##   
## Done.



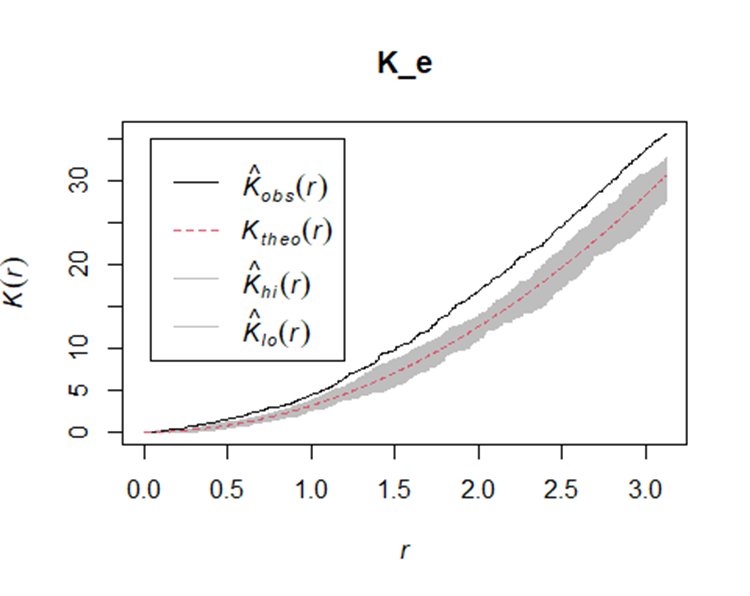
Q8. What does the plot show us?

The plot displays the results of a spatial analysis using Ripley's G-function, with 99 simulations to create an envelope under the assumption of Complete Spatial Randomness (CSR). The solid line (`G\_obs(r)`) represents the observed G-function for 'hit' data, while the dashed line (`G\_theo(r)`) denotes the theoretical expectation under CSR. The shaded area represents the range of G-functions from the simulations. The observed G-function lying outside the CSR envelope suggests significant spatial structure in the 'hit' data, with the pattern deviating from randomness, potentially indicating clustering or dispersion depending on whether the observed line is above or below the envelope.

K\_e <- envelope(rob, Kest, nsim=19, nrank=1)  
plot(K\_e)

#### Result

## Generating 19 simulations of CSR  
## 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,   
## 19.  
##   
## Done.



Q9. describe the pattern.

This plot represents a spatial point pattern analysis using Ripley's K-function. The solid black line (`K\_obs(r)`) shows the observed K-function for the data, while the dashed red line (`K\_theo(r)`) indicates the theoretical K-function under complete spatial randomness (CSR). The grey envelope (`K\_hi(r)` and `K\_lo(r)`) is generated from 19 simulations under CSR. Observing that the `K\_obs(r)` line lies within the envelope for the most part suggests that the pattern of the data does not significantly deviate from CSR. This implies that the spatial distribution of the points in the dataset is essentially **random**, with no significant evidence of clustering or regularity at the scales represented by `r`.