*Spatial Statistics Lab 4*

**Onyedikachi J Okeke**

2024-03-11

### **0.0 To 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 Consider using ‘sf’ or ‘terra’ instead

library(sp)  
library(spatstat)  
library(sf)  
library(spatstat.geom)  
library(ctv)  
library(terra)  
library(spdep)  
library(terra)  
library(RColorBrewer)  
library(classInt)  
library(epitools)  
library(DCluster)  
library(rgdal)  
library(maptools)

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

## terra 1.7.65

##   
## Attaching package: 'terra'

## The following objects are masked from 'package:spatstat.geom':  
##   
## area, delaunay, is.empty, rescale, rotate, shift, where.max,  
## where.min

## Loading required package: spData

## Loading required package: boot

##   
## Attaching package: 'boot'

## The following object is masked from 'package:spatstat.explore':  
##   
## envelope

## Loading required package: MASS

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:terra':  
##   
## area

## The following object is masked from 'package:spatstat.geom':  
##   
## area

## Please note that rgdal will be retired during October 2023,  
## plan transition to sf/stars/terra functions using GDAL and PROJ  
## at your earliest convenience.  
## See https://r-spatial.org/r/2023/05/15/evolution4.html and https://github.com/r-spatial/evolution  
## rgdal: version: 1.6-7, (SVN revision 1203)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 3.6.2, released 2023/01/02  
## Path to GDAL shared files: C:/Users/GIS/AppData/Local/R/win-library/4.3/rgdal/gdal  
## GDAL does not use iconv for recoding strings.  
## GDAL binary built with GEOS: TRUE   
## Loaded PROJ runtime: Rel. 9.2.0, March 1st, 2023, [PJ\_VERSION: 920]  
## Path to PROJ shared files: C:\Program Files\PostgreSQL\14\share\contrib\postgis-3.2\proj  
## PROJ CDN enabled: FALSE  
## Linking to sp version:2.1-0  
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,  
## use options("rgdal\_show\_exportToProj4\_warnings"="none") before loading sp or rgdal.

##   
## Attaching package: 'rgdal'

## The following object is masked from 'package:terra':  
##   
## project

## Please note that 'maptools' will be retired during October 2023,  
## plan transition at your earliest convenience (see  
## https://r-spatial.org/r/2023/05/15/evolution4.html and earlier blogs  
## for guidance);some functionality will be moved to 'sp'.  
## Checking rgeos availability: FALSE

##   
## Attaching package: 'maptools'

## The following objects are masked from 'package:DCluster':  
##   
## readSplus, sp2WB

## The following object is masked from 'package:sp':  
##   
## sp2Mondrian

### **Loading city\_limits\_km and StLouisCrime2014**

library(rgdal)  
library(spatstat)  
library(sp)  
library(sf)  
library(spatstat.geom)  
library(maptools)  
  
# Read the shapefile  
S <- readOGR("C:/Spatial Statistics Labwork/Lab4Data/city\_limits\_km.shp")  
  
# Convert the Spatial object to SpatialPolygons  
SP <- as(S, "SpatialPolygons")  
  
SP <- as(S, "SpatialPolygons")  
  
W <- as(SP, "owin")  
  
plot(W)  
  
xy <- read.table("C:/Spatial Statistics Labwork/Lab4Data/StLouisCrime2014.txt", header=T, sep="\t")  
attach(xy)  
pp <- ppp(X, Y, window=W, marks=CRIME)  
  
plot(pp)  
  
gun <- pp[CRIME=="DISORDERLY"]  
rob <- pp[CRIME=="BURGLARY"]  
hit <- pp[CRIME=="HITANDRUN"]  
  
plot(density(gun))  
contour(density(gun), add=T)  
plot(gun, add=T)  
  
#Kernel density visualization is performed in spatstat using the density() function  
  
plot(density(gun, 0.25))  
  
d250 <- density(gun, 0.25)  
plot(d250)  
contour(d250, add=T)  
plot(gun, add=T)  
  
plot(gun, add=T)  
r <- bw.diggle(gun)  
  
r

##### Result

## Warning: OGR support is provided by the sf and terra packages among others

## Warning: OGR support is provided by the sf and terra packages among others

## Warning: OGR support is provided by the sf and terra packages among others

## Warning: OGR support is provided by the sf and terra packages among others

## Warning: OGR support is provided by the sf and terra packages among others

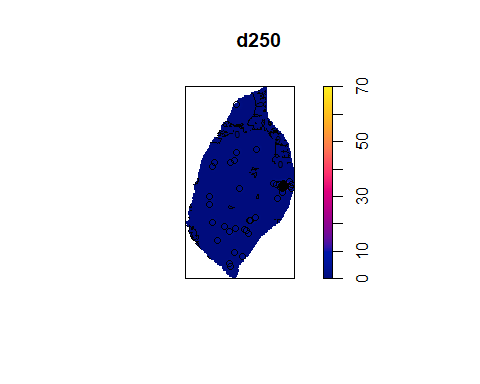
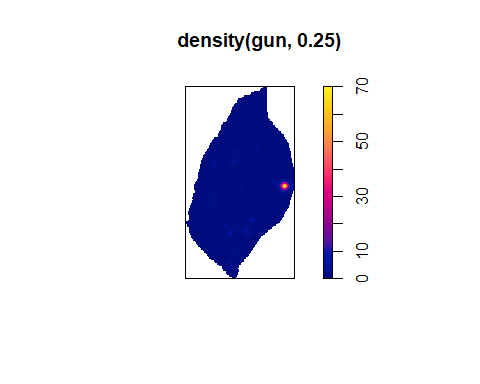
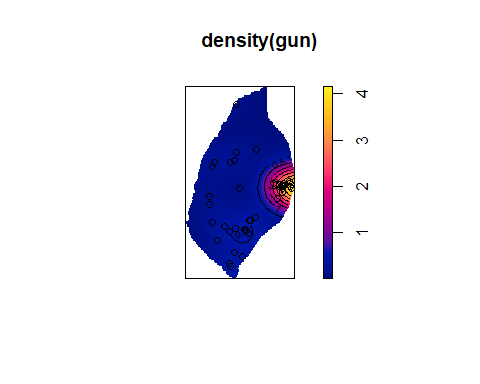
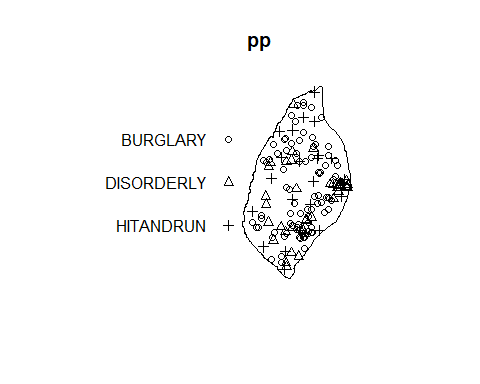
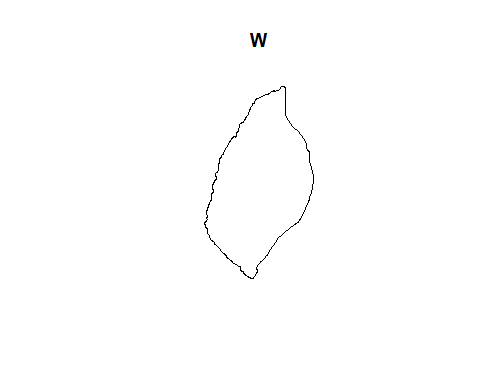
## Warning: OGR support is provided by the sf and terra packages among others

## Warning: OGR support is provided by the sf and terra packages among others

## OGR data source with driver: ESRI Shapefile   
## Source: "C:\Spatial Statistics Labwork\Lab4Data\city\_limits\_km.shp", layer: "city\_limits\_km"  
## with 1 features  
## It has 3 fields

## Warning: data contain duplicated points

## sigma   
## 0.003054447



Q4. Create density maps (in R) of the gun homicide data, experimenting with different kernel density bandwidths. Provide a commentary discussing the most suitable bandwidth choice for this analysis visualization method. (15 points)

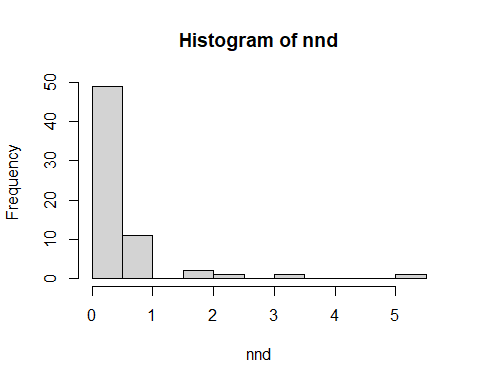
## **convert it to a spatstat point pattern object, with the different crime types as an identifying mark**

nnd <- nndist.ppp(gun)  
hist(nnd)  
summary(nnd)  
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 Quadrat analysis**

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

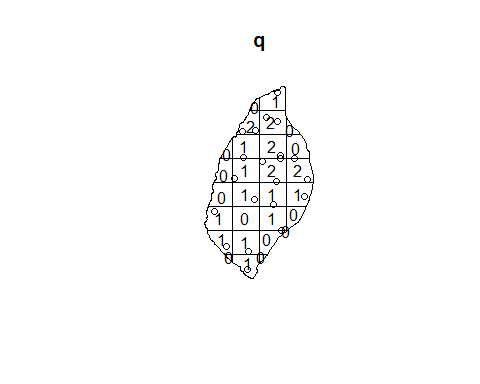
##   
## Chi-squared test of CSR using quadrat counts  
##   
## data: hit  
## X2 = 11.309, df = 27, p-value = 0.006984  
## alternative hypothesis: two.sided  
##   
## Quadrats: 28 tiles (irregular windows)

## 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. Take a screenshot of the plotted figure. Is it a clustered, regular, or random pattern?

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

library(spatstat)   
  
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)

#### Result

