SPA5001 Assignment

Matthew Finster
Student ID: 21702717

May 2024

'This is my own work. I have not copied any of it from anyone else.' - Matthew Finster, 25/05/2024

Contents

1 Question 1. Simulating a realisation of the inhomogeneous Poisson process

A realisation of an inhomogeneous Poisson process in the interval space of $[0, 10]^2$ was simulated. In this simulation:

- The points form two groups within the interval space;
- The centres of the groups have random locations; and
- For each group, all distances between the points and the centre are less than 2.

The plots for the two simulations of this realisation can be found in Figure 1 below.

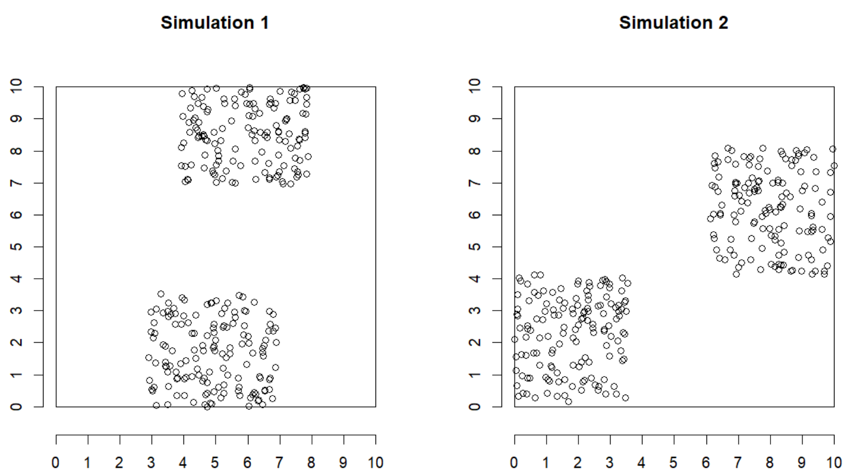

Figure 1: Simulations of realisation of the inhomogeneous Poisson process

The R code for this realisation is below:

```
# Installing the spatspat library
library(spatstat)
# Setting up the plot layout (1 row, 2 columns)
par(mfrow=c(1,2))
# Creating four random numbers within the interval space which will be used as the two centres [x1,y1] and [x2, y2]
RandomCentres <-runif(4,min = 0, max = 10)
RandomCentres
```

```
[1] 5.886135 8.969012 4.905167 1.586330
# Setting up the intensity function within the inhomogeneous Poisson process whereby 10 points are within a 2 unit distance from the centre using numbers 1 and 2 as x1 and y1
lambda1 <- function(x, y){
  10 * as.numeric((abs(x -RandomCentres[1])<2) & (abs(y -RandomCentres[2])<2))}
# Plotting the first realisation
plot(rpoispp(lambda1, win=owin(c(0,10),c(0,10))),
     main = "Figure 1. Simulation 1")
     axis(1, at = seq(0, 10, by = 1))
     axis(2, at = seq(0, 10, by = 1))
# Setting up intensity function for the second group using numbers 3 and 4 as x2 and y2
lambda2 <- function(x, y){
  10 * as.numeric((abs(x -RandomCentres[3])<2) & (abs(y -RandomCentres[4])<2))}

# Plotting the second realisation and overlaying it on the first plot
plot(rpoispp(lambda2, win=owin(c(0,10),c(0,10))),add=TRUE)

# Running the code again for the second simulation
RandomCentres <-runif(4,min = 0, max = 10)
RandomCentres
[1] 8.131544 6.126712 1.554120 2.157995
lambda1 <- function(x, y){
  10 * as.numeric((abs(x -RandomCentres[1])<2) & (abs(y -RandomCentres[2])<2))}
plot(rpoispp(lambda1, win=owin(c(0,10),c(0,10))),
     main = "Figure 2. Simulation 2")
axis(1, at = seq(0, 10, by = 1))
axis(2, at = seq(0, 10, by = 1))
lambda2 <- function(x, y){
  10 * as.numeric((abs(x -RandomCentres[3])<2) & (abs(y -RandomCentres[4])<2))}
plot(rpoispp(lambda2, win=owin(c(0,10),c(0,10))),add=TRUE)
```

2 Question 2. Investigating data of a rat's gastric mucosa

The data **mucosa** from SPATSTAT was used to investigate trends within a rat's gastric muscosa. The data give the locations of the centres of two types of cells in a cross-section of the gastric mucosa of a rat. This is a marked point pattern. The marks have the levels 'ECL' (enterochromaffin) and 'other'.

a. A plot of the data was created, which can be seen in Figure 2. The R code to generate this plot can be found below Figure 2.

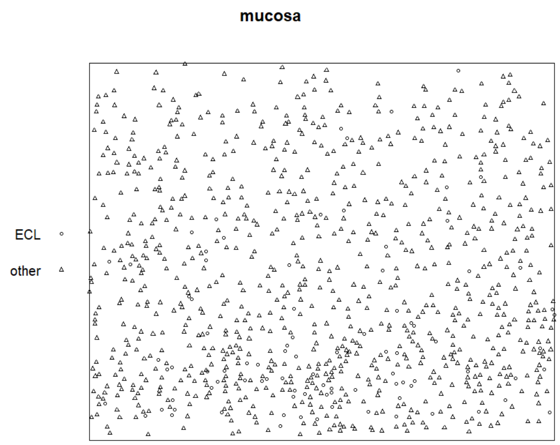

Figure 2: Plot of mucosa data

```
## Question 2 Investigating data of a rat's gastric mucosa
# Accessing and plotting the mucosa data
data(mucosa)
plot(mucosa, cex = 0.5)
```

b. The intensity of the dataset was investigated and estimated to be *1191.358 points per square unit*. However, this constant intensity is unlikely to be a good estimate, as displayed in the uneven distribution of mucosa in Figure 2, with more data points towards the lower values on the x and y axes. The hypothesis that this is stationary data is unlikely to be true. Given this, intensity of points changing from location to location was investigated using quadrant counting and kernel smoothing using density plots. These were plotted in Figure 3. The R code to generate these plots can be found below Figure 3.

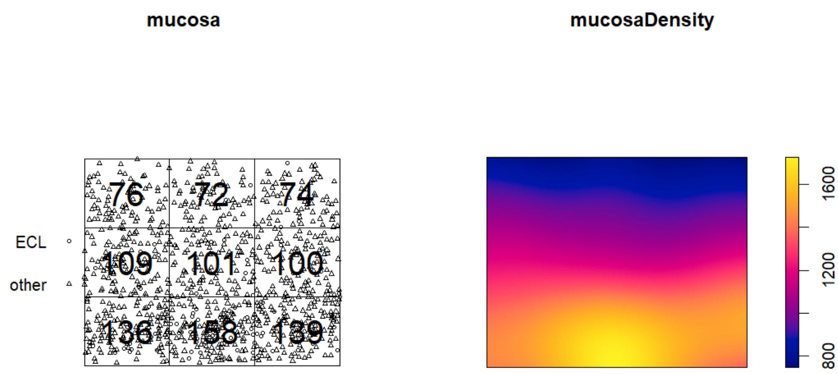

Figure 3: Plots of mucosa intensity

```
# Investigating the constant intensity of the dataset
summary(mucosa)$intensity
[1] 1191.358
# Investigating the non-stationary intensity of the dataset
quadratcount(mucosa, nx = 3, ny = 3)
                   x
y              [0,0.333) [0.333,0.667) [0.667,1]
[0.54,0.81]         76             72         74
[0.27,0.54)        109            101        100
[0,0.27)           136            158        139
Q <- quadratcount(mucosa, nx = 3, ny = 3)
# Plotting the quadrant over the original mucosa plot
plot(Q, add = TRUE, cex = 2)
# Plotting density
mucosaDensity <- density(mucosa, sigma = 0.15)
plot(mucosaDensity)
par(mfrow=c(1,2))
```

c. The actual locations of the mucosa cells and a contour plot (using the intensity estimated in *Question 2b* via the kernel smoothing method) were plotted together in Figure 4. The R code to generate this plot can be found below Figure 4.

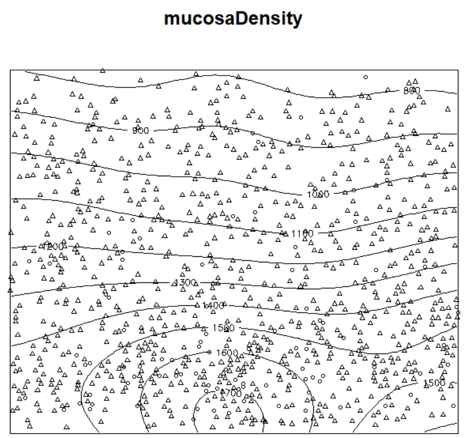

Figure 4: Plot of mucosa density

```r
# Plotting density
contour(mucosaDensity)
#Adding data points over the top of contours
plot(mucosa, add = TRUE, cex = 0.5)
```

d. A Poisson process (using the estimated intensity from *Question 2b* and *Question 2c* via the kernel smoothing method) was simulated in Figure 5 below. The R code to create this plot is below:

```r
# Simulating the inhomogeneous Poisson process
lambda <- density(mucosa)
mucosaPoissonInhomogeneous <- rpoispp(lambda)
plot(mucosaPoissonInhomogeneous)
```

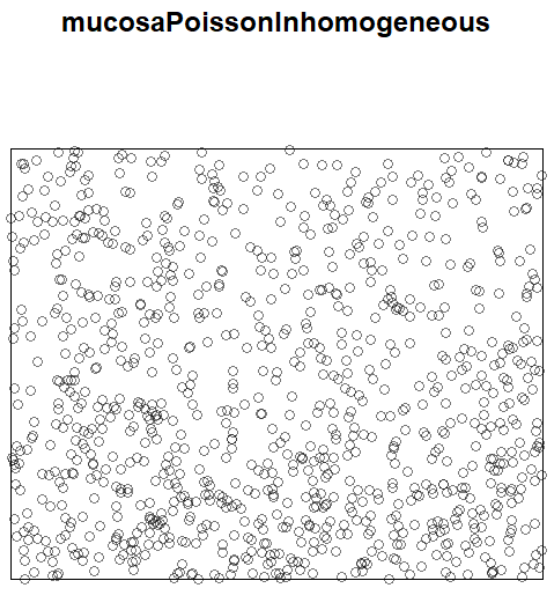

Figure 5: Simulated inhomogeneous Poisson process with the estimated intensity

e. The data was separated into the sub-patterns of points of types 'ECL' and 'other'. Their intensities were estimated again using kernel smoothing. These estimates were plotted in Figure 6. The R code used to generate these intensity plots can be found beneath Figure 6, although it is now understood that the following would have been just as effective:

```
plot(density(split(mucosa)))
```

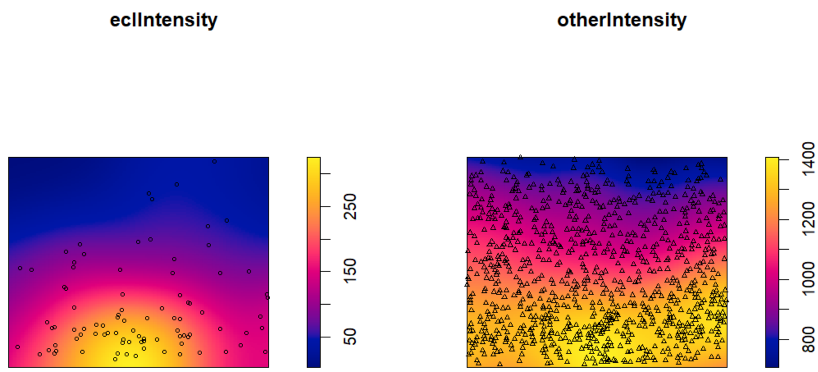

Figure 6: Intensity of 'ECL' data points and of 'other' data points

```
#Understanding which marks relate to which subset
mucosa$marks
# Subsetting the dataset
eclSubset <- mucosa[1:89]
otherSubset <- mucosa[90:mucosa$n]
# Ensuring the dataset was subsetted correctly
eclSubset$marks
eclSubset$n
class(eclSubset)
otherSubset$marks
otherSubset$n
class(otherSubset)

# Plotting ECL intensity
par(mfrow=c(1,2))
eclIntensity <- density(eclSubset, sigma = 0.15)
plot(eclIntensity)
plot(eclSubset, add = TRUE, cex = 0.5)

# ¨Plotting other intensity
otherIntensity <- density(otherSubset, sigma = 0.15)
plot(otherIntensity)
plot(otherSubset, add = TRUE, cex = 0.5)
```

f. Based on the intensity plots generated in Figure 6, it is not unreasonable to imagine that the location of *'ECL'* cells and *'other'* type of cells are correlated. The cross-type pair correlation function for *'ECL'* and *'other'* marks were plotted in Figure 7. The horizontal green line at $y = 1$ represents the expected value under complete spatial randomness. Figure 7 shows that there is a strong inhibition between *'ECL'* and *'other'* cells at very short distances, while at all other distances the two categories of cells appear to be clustered with respect to one another - although this attraction is relatively weak. The code for this plot can be found beneath Figure 7.

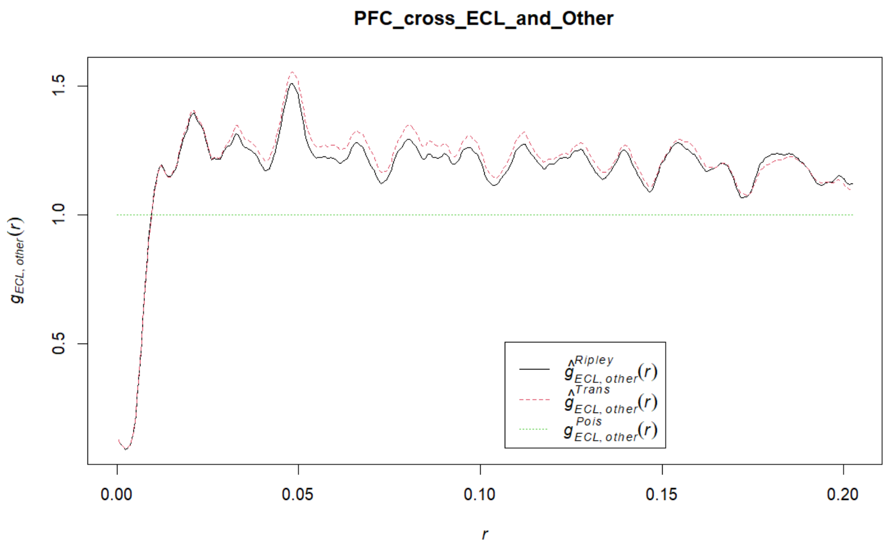

Figure 7: Cross-type pair correlation function for a multitype point pattern between 'ECL' and 'other'

```r
## Plotting the cross-type pair correlation function for 'ECL' and 'other'
PFC_cross_ECL_and_Other <- pcfcross(mucosa, "ECL", "other")
plot(PFC_cross_ECL_and_Other)
```

g. The marks were removed from the point pattern. The uniform distribution of the x coordinate and of the y coordinate were tested using spatial Kolmogorov-Smirnov tests. The results can be found in Figure 8 below. The results indicate that the observed distribution of the x coordinate and expected distribution of complete spatial randomness (CSR) strongly align and thus, along with the *p-value* of 0.734, the hypothesis of CSR is not rejected by this test. In contrast, the results indicate that the observed distribution of the y coordinate do not align with the expected distribution of CSR and thus, along with the the *p-value* of < 0.001, the hypothesis of CSR is rejected by the test. These test results are logical when considering how the intensity plots show no visible correlation along the *x-axis* but a visibly strong trend along the *y-axis*.

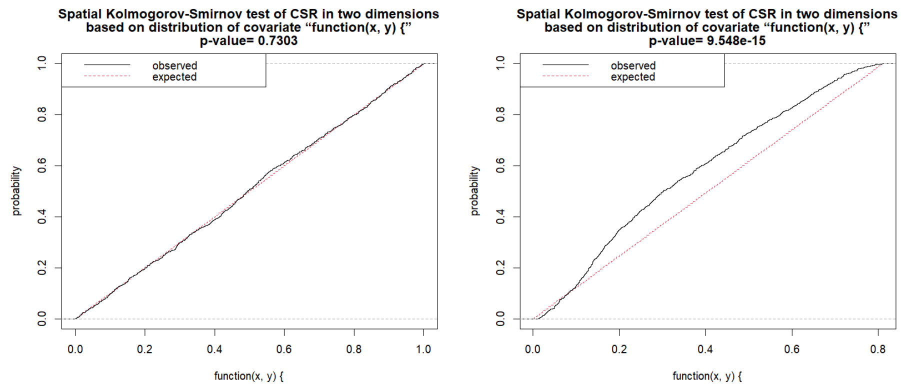

Figure 8: Spatial Kolmogorov-Smirnov tests for the uniform distributions of the x and coordinates

```
# Removing marks from the point pattern
mucosaNoMarks <- unmark(mucosa)
par(mfrow=c(1,2))
# Testing the uniform distribution of the x coordinate
testx <- cdf.test(mucosaNoMarks, function(x, y) {x})
testx$p.value
[1] 0.7338408
plot(testx)
#Testing the uniform distribution of the y coordinate
testy <- cdf.test(mucosaNoMarks, function(x, y) {y})
testy$p.value
[1] 8.992806e-15
plot(testy)
```

3 Question 3. Investigating data from New Brunswick Department of Natural Resources

The data **nbfires** from SPATSTAT was used to investigate trends of all fires falling under the jurisdiction of New Brunswick Department of Natural Resources from 1987 to 2003 inclusive.

a. There were 7108 fires recorded between 1987 and 2003 by NBDNR. Figure 9 shows that most fires were forest fires, accounting for 65.1 per cent ($n = 4627$) of all fires. Grass fires accounted for 12.77 per cent ($n = 908$), dump fires accounted for 11.75 per cent ($n = 835$), while fires of all 'other' types accounted for the final 10.38 per cent ($n = 738$). A plot showing the distribution of fire types was produced in Figure 9. The R code used to generate this plot can be found below Figure 9.

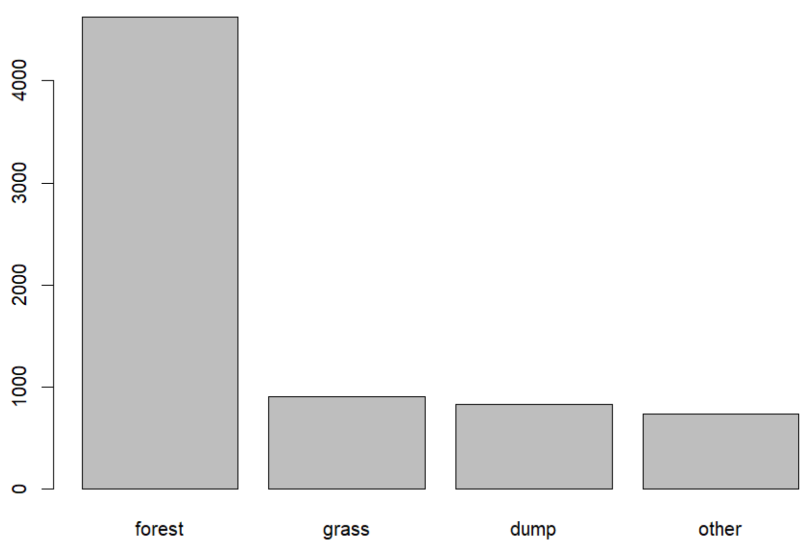

Figure 9: Types of fires

```
# Assignment Question 3 Investigating data from New Brunswick Department of Natural Resources
# Accessing and plotting the nbfires data
data(nbfires)
summary(nbfires$marks$fire.type)
forest  grass   dump  other
```

```
  4627     908     835     738
plot(nbfires$marks$fire.type)
```

b. Density plots of each fire type were produced with their corresponding points overlaid, which can be seen in Figure 10. The R code used to generate these plots can be found below Figure 10.

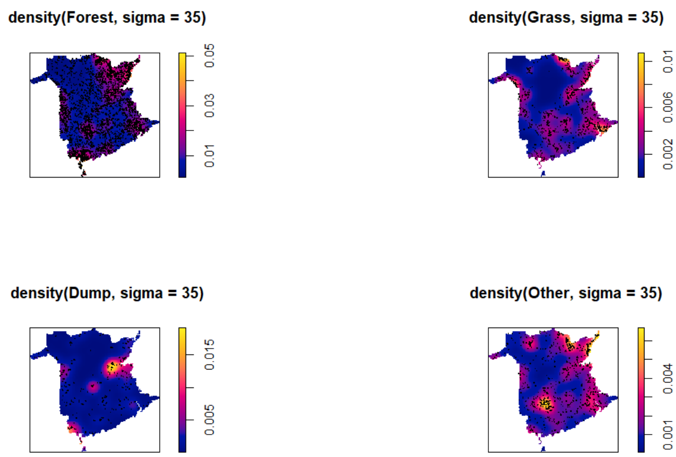

Figure 10: Density plots for each fire type

```
# Manually extracting the data
Forest <- nbfires[nbfires$marks$fire.type == "forest", ]
Grass <- nbfires[nbfires$marks$fire.type == "grass", ]
Dump <- nbfires[nbfires$marks$fire.type == "dump", ]
Other <- nbfires[nbfires$marks$fire.type == "other", ]
# Plotting densities with points overlayed
par(mfrow = c(2, 2))
plot(density(Forest, sigma = 35))
plot(Forest, add = TRUE, cex = 0.1)
plot(density(Grass, sigma = 35))
plot(Grass, add = TRUE, cex = 0.1)
plot(density(Dump, sigma = 35))
plot(Dump, add = TRUE, cex = 0.1)
plot(density(Other, sigma = 35))
plot(Other, add = TRUE, cex = 0.1)

```

```
# Note: the following function does not split via fire type and specifying nbfires£marks£fire.type makes the command unusable
# plot(density(split(nbfires)))
```

c. The image of empty space distances for the locations of forest fires was plotted, which can be seen in Figure 11. The R code to generate this plot can be found below Figure 11.

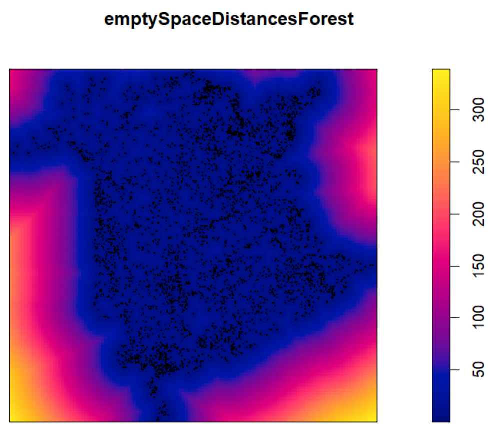

Figure 11: Image of empty space distances for the locations of forest fires

```
par(mfrow = c(1, 1))
# Plotting the image of empty space distances for the locations of forest fires
emptySpaceDistancesForest <- distmap(Forest)
plot(emptySpaceDistancesForest)
plot(Forest, add = TRUE, cex = 0.1)
```

d. The **F** function for the locations of forest fires was computed and plotted, which can be seen in Figure 12. $\hat{F}(r)$ values are defined from the distribution function of the observed empty space distances. Given that the $\hat{F}(r)$ values are smaller than $F_{\mathrm{POI}}(r)$ values in Figure 12, this indicates that empty space distances in the point pattern are longer than what they would be for a regularly spaced Poisson process, suggesting a clustered pattern somewhere in the data. The code for computing and plotting this **F** function can be found below Figure 12.

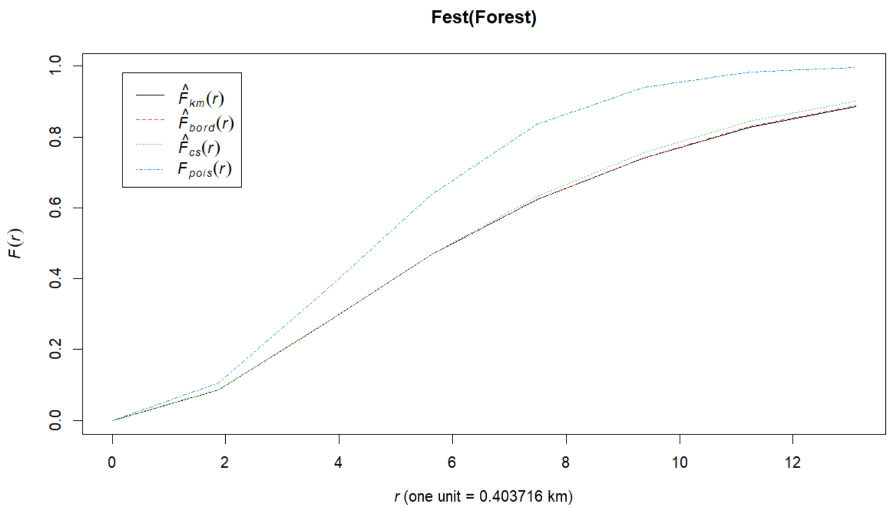

Figure 12: F function for the locations of forest fires

```
# Computing and plotting the F function for the locations
↪   of forest fires
plot(Fest(Forest))
# Computing and plotting the G function for the locations
↪   of forest fires
plot(Gest(Forest))
```

e. The **G** function for the locations of forest fires was computed and plotted, which can be seen in Figure 13. $\hat{G}(r)$ values are defined from the distribution function of the nearest neighbour distances. Given that $\hat{G}(r)$ values are mostly greater than $F_{\mathrm{POI}}(r)$ values in Figure 13, this indicates that nearest neighbour distances are shorter than for a Poisson process, suggesting a clustered pattern in the data. The code for computing and plotting this **G** function can be found below Figure 12.

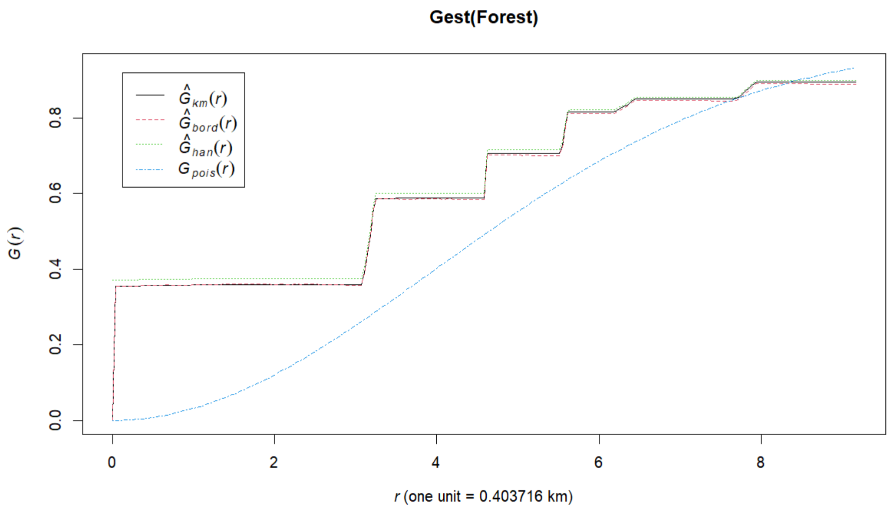

Figure 13: G function for the locations of forest fires

f. All marks were removed. An inhomogeneous Poisson model with an intensity that is a polynomial of degree 3 of x and y coordinates was fitted. The locations from the fitted trend are plotted as circles while the actual data locations are plotted as + symbols in the same image, which can be seen in Figure 14. It is evident that the locations from the fitted model and the actual locations closely align. The R code used to generate this figure can be found below.

```r
# Consider only forest fires and remove all marks.
forestNoMarks <- unmark(Forest)
# Fit an inhomogeneous Poisson model with an intensity that
# is a polynomial of degree 3 of x and y coordinates.
# Plot the corresponding trend and data locations
# in the same image.
fit1<-ppm(forestNoMarks, ~polynom(x, y, 3))
plot(fit1, how = "image", se = FALSE, main = "Fitted
  inhomogeneous Poisson model with data points")
plot(forestNoMarks, add = TRUE, pch = 3, cex = 0.5)
```

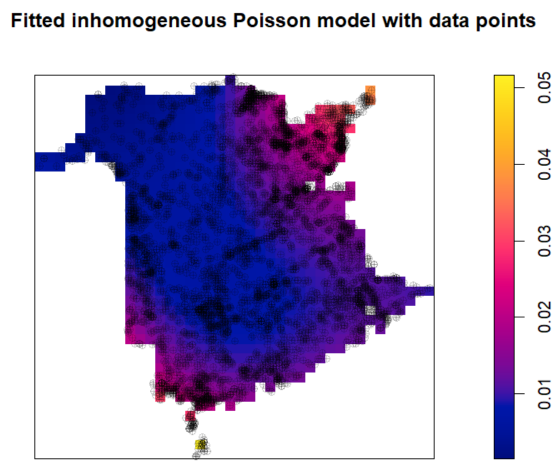

Figure 14: Fitted inhomogeneous Poisson model locations *(*O*)* with actual data locations *(+)*

4 Question 4. Studying the Poisson process with the intensity

$$\lambda(x, y, t) = 10\,000 \times (1 - x)^2 \times y^2 \times t^3, \qquad x, y, t \in [0, 1]$$

a. This inhomogeneous Poisson process was simulated. The R code for this Poisson process can be found below.

```
## Assignment Question 4
library(stpp)
# (x, y, t) = 10 000  (1 - x)^2  y^2  t^3, x
# Simulate the Poisson process.
q4lambda <- function(x, y, t) {10000 * (1-x)^2 * y^2 * t^3}
ipp <- rpp(lambda = q4lambda)
```

b. A static display of the result can be seen in Figure 15. The R code to produce this static display can be found below Figure 15.

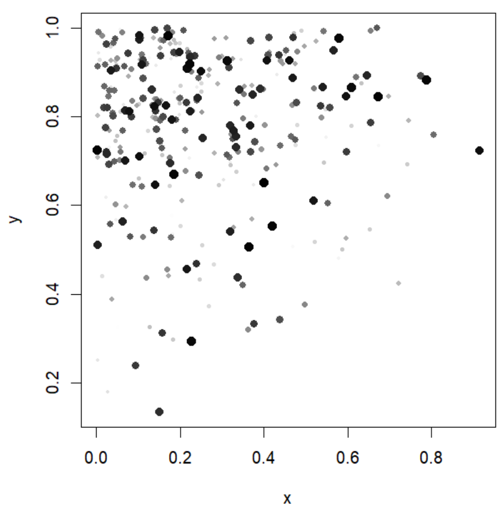

Figure 15: Static display of the simulated inhomogenous Poisson process with the intensity $\lambda(x, y, t) = 10\,000 \times (1 - x)^2 \times y^2 \times t^3, \quad x, y, t \in [0, 1]$

```
# Giving a static display of the result.
ipp2 <- as.3dpoints(ipp$xyt)
plot(ipp2, pch=19, mark=TRUE)
```

c. An animation of the process was run. The R code used to produce this animation can be found below.

   It was observed during the animation that most points were added near the end of runtime (as t increased). This is to be expected as in the intensity formula, t is cubed. Thus, when t is small, t^3 is very small, leading to fewer points being added. As t approaches 1, t^3 increases rapidly because it is raised to the power of 3, significantly increasing the intensity and thus the number of points being added.

   It was also observed in the animation (but can also be observed in Figure 15) that points were more likely to be added when x is small and when y was large due to the $(1-x)^2$ and y^2 terms, respectively.

```r
# Animating the process
animation(ipp$xyt,runtime = 5)
```

d. A two-panel plot showing spatial locations and cumulative times was produced, as can be seen in Figure 16. The R code used to produce this two-panel plot can be found below.

   The xy locations plot in Figure 16 provides further evidence that points are more likely to be added when x is small due to the $(1-x)^2$ term. The xy locations plot also provides further evidence that points are more likely to be added when y is large due to the y^2 terms.

   The *cumulative number* plot in Figure 16 also provides further evidence that points are more likely to be added toward the end of runtime (when t is larger) due to the t^3 term. As t is closer to 0, less points are added. As t begins to rise above 0.5, the number of points rapidly increases, due to t being raised to the power of 3.

```r
# Producing a two-panel plot showing spatial locations and
↪  cumulative times
plot(ipp$xyt)
```

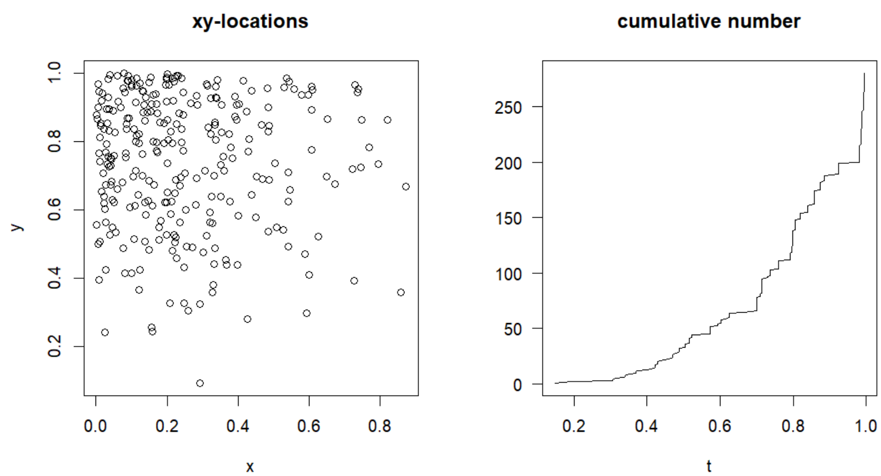

Figure 16: Two-panel plot showing spatial locations and cumulative times for the inhomogeneous Poisson process with the intensity $\lambda(x, y, t) = 10\,000 \times (1 - x)^2 \times y^2 \times t^3, \quad x, y, t \in [0, 1]$

e. Lastly, the temporal intensity functions of the inhomogeneous Poisson process with the intensity $\lambda(x, y, t) = 10\,000 \times (1-x)^2 \times y^2 \times t^3, \quad x, y, t \in [0, 1]$ were estimated and plotted in Figure 17. The R code to produce the plot can be found below.

```
par(mfrow = c(1, 1))
# Estimating and plotting the temporal intensity functions
↪ of the process
plot(density(ipp$xyt[,3]), main = "Temporal intensity
↪ functions of the process")
```

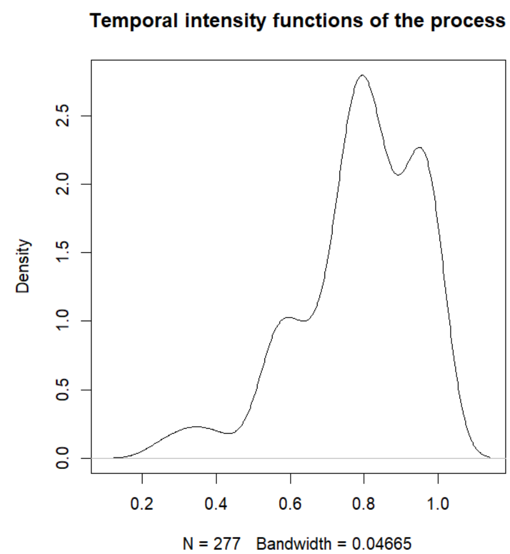

Figure 17: Temporal intensity functions of the inhomogeneous Poisson process with the intensity $\lambda(x, y, t) = 10\,000 \times (1 - x)^2 \times y^2 \times t^3, \quad x, y, t \in [0, 1]$

5 Question 5. Investigating data from a novel dataset

a. A novel dataset was found to investigate. This dataset provides the geographical locations of all 74 public toilets in the City of Melbourne area. This is interesting data to examine as a potential infrastructural solution to combat the lingering effects of social isolation during the COVID-19 epidemic. Public toilets can offer infrastructural support for those looking to organise social opportunities with others, and thus can positively impact social health outcomes. This data's URL address is

   https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/public-toilets/exports/csv.

   The first 5 rows can be found in the R code below:

```
## Assignment Question 5
setwd("C:\\Users\\anais\\Documents\\Matt\\Etudies\\9. Spatial Analysis\\Assignment")
# Importing the data in R
melbourneToilets <- read.csv("public-toilets.csv", header=TRUE)
# Cleaning the data and printing the first 5 rows
nrow(melbourneToilets)
[1] 74
melbourneToilets <- melbourneToilets[,7:8]
head(melbourneToilets)
        lat      lon
1 -37.80399 144.9591
2 -37.81384 144.9631
3 -37.80464 144.9629
4 -37.78622 144.9550
5 -37.79373 144.9304
6 -37.81528 144.9776
nrow(melbourneToilets)
[1] 74
```

b. The spatial intensity of the data was investigated and plotted in Figure 18 below. It can be noted that the spatial intensity of Melbourne toilets is small in the bottom left quadrants, which could be due to the ocean (Port Phillip Bay). The R code to generate these plots can be found below Figure 18.

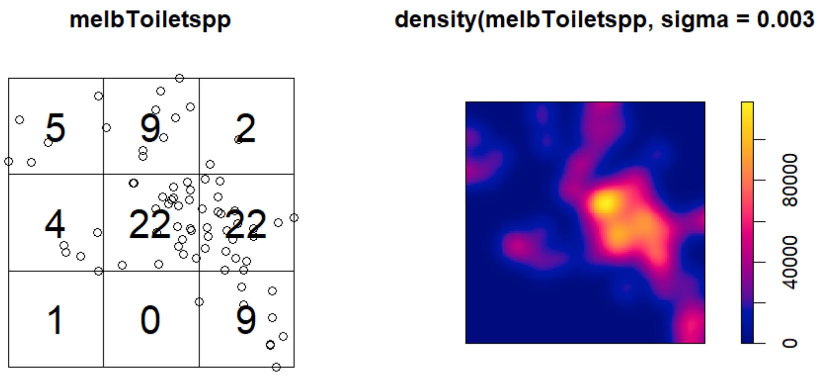

Figure 18: Intensity plots of Melbourne public toilets

```
par(mfrow = c(1, 2))
# Creating a ppp object & plotting
library(spatstat)
melbToiletspp <- ppp(melbourneToilets$lon, melbourneToilets$lat, window=owin(xrange=c(min(melbourneToilets$lon), max(melbourneToilets$lon)), yrange=c(min(melbourneToilets$lat), max(melbourneToilets$lat))))
# 2x2 plots
par(mfrow=c(1,2))
# Plotting Melbourne public toilet data points
plot(melbToiletspp)
# Applying quadrant count
quadratMelbToiletspp <- quadratcount(melbToiletspp, nx = 3, ny = 3)
plot(quadratMelbToiletspp, add = TRUE, cex = 2)
# Creating a density plot
plot(density(melbToiletspp, sigma = 0.003))
```

c. A plot of the actual public toilet locations in Melbourne was placed over the intensity estimated from *Question 5b* using the kernel smoothing method. This can be seen in Figure 19. The R code to generate this plot can be found below Figure 19.

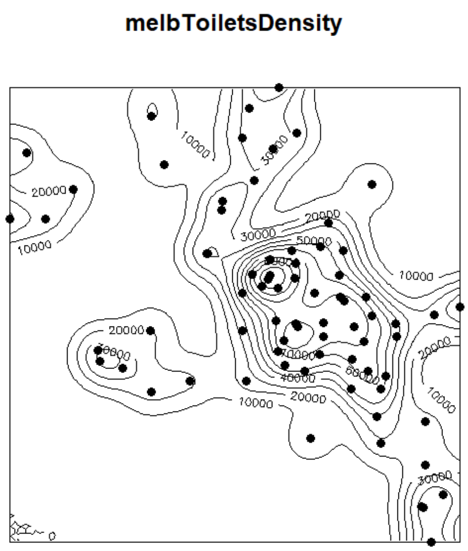

Figure 19: Contour plot of estimated intensity of public toilets in the City of Melbourne along with actual data points

```r
# Contour plot
par(mfrow=c(1,1))
melbToiletsDensity <- (density(melbToiletspp, sigma = 0.003))
contour(melbToiletsDensity)
plot(melbToiletspp, add = TRUE, pch = 16)
```

d. A Poisson process (using the estimated intensity from *Question 5b* and *Question 5c*) of public toilets in the City of Melbourne was simulated and plotted in Figure 20. The R code used to generate this plot can be found below Figure 20.

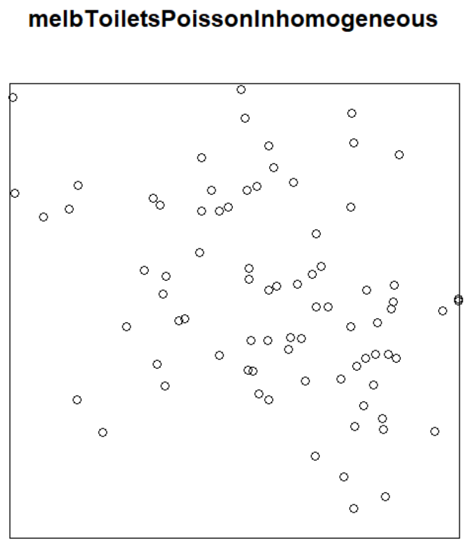

Figure 20: Simulated Poisson process with estimated intensity for public toilets in the City of Melbourne

```
## Poisson process with estimated intensity
# Inhomogeneous
lambdaMelbToilets <- density(melbToiletspp)
melbToiletsPoissonInhomogeneous <-
    rpoispp(lambdaMelbToilets)
plot(melbToiletsPoissonInhomogeneous)
```

e. The estimated intensity from *Question 5b* and *Question 5c* and the simulated Poisson process from *Question 5d* were plotted in the same image. These can be found in Figure 21. The R code to produce these plots can be found below Figure 21.

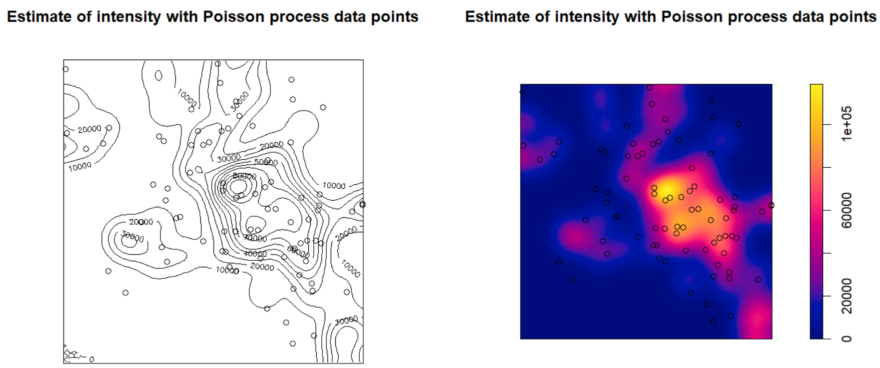

Figure 21: Estimated intensity with data points plotted from Poisson process

```
par(mfrow=c(1,2))
# Plotting estimated intensity and the simulated process in
↪   the same image
melbToiletsDensity <- (density(melbToiletspp, sigma =
↪   0.003))
contour(melbToiletsDensity, main = "Estimate of intensity
↪   with Poisson process data points")
plot(melbToiletsPoissonInhomogeneous, add = TRUE)

plot(melbToiletsDensity, main = "Estimate of intensity with
↪   Poisson process data points")
plot(melbToiletsPoissonInhomogeneous, add = TRUE)
```

Author: Matthew Finster
Student ID: 21702717