

## Libraries

```
library(spatstat)

## Loading required package: spatstat.data
## Loading required package: spatstat.geom
## spatstat.geom 3.2-1
## Loading required package: spatstat.random
## spatstat.random 3.1-5
## Loading required package: spatstat.explore
## Loading required package: nlme
## spatstat.explore 3.2-1
## Loading required package: spatstat.model
## Loading required package: rpart
## spatstat.model 3.2-4
## Loading required package: spatstat.linnet
## spatstat.linnet 3.1-1

##
## spatstat 3.0-6
## For an introduction to spatstat, type 'beginner'
```

```
library(reticulate)
library(RColorBrewer)
pd <- import("pandas")
```

## Definition of input Data

```
chains_to_investigate <- list("Chain 1" )
```

```
scenario <- 1
```

## Window

```
window = owin(c(0,1000), c(0,1000))
```

## Outbreak

Outbreak artificially made to test model:

```
ppp_outbreak <- ppp(x = c(750, 150, 50, 450, 50, 850, 850, 950, 50, 750), y = c(150, 250, 250, 450, 850, 150, 250, 250, 450, 850),  
ppp_outbreak <- rescale(ppp_outbreak, 1000, "km")
```

## Population Data

uniform population of 5 in each cell (500 total)

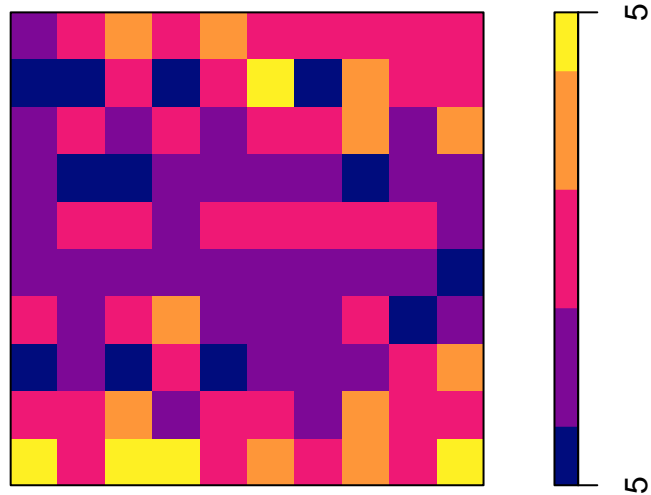
```
no_of_cells <- 100  
population_per_cell <- 5  
  
# initialize an empty data.frame  
df_population <- data.frame("population" = numeric(), "x_centroid" = numeric(), "y_centroid" = numeric())  
  
# set values  
y <- -50  
x <- 50  
  
for (i in 1:no_of_cells) {  
  if (i %% 10 == 1) {  
    y <- y + 100  
    x <- 50  
  }  
  
  df_population[i, "y_centroid"] <- y  
  df_population[i, "x_centroid"] <- x  
  
  x <- x + 100  
}  
  
df_population$population <- rep(population_per_cell, nrow(df_population))
```

```
ppp_population <- ppp(x = df_population$x_centroid, y = df_population$y_centroid, window = window, marks = marks(ppp_outbreak))  
ppp_population <- rescale(ppp_population, 1000, "km")  
smo_population <- density(ppp_population, eps = 0.1, positive = TRUE, weights = marks(ppp_population))
```

```
im_population <- smo_population  
im_population <- eval.im(im_population / 100)  
im_population <- eval.im(pmax(im_population, 1e-10))
```

```
plot(im_population)
```

## im\_population



## Quadrature Scheme

```
Q <- quadscheme(ppp_outbreak, eps = 0.1)
```

## Null Model

```
fit0 <- ppm(Q ~ offset(log(im_population)))
print(fit0)
```

```
## Nonstationary Poisson process
## Fitted to point pattern dataset 'Q'
##
## Log intensity: ~offset(log(im_population))
##
## Fitted trend coefficient: (Intercept) = 0.6931472
##
##           Estimate      S.E.    CI95.lo CI95.hi Ztest      Zval
## (Intercept) 0.6931472 0.3162278 0.07335215 1.312942      * 2.191924
```

$$\lambda_0(u) = e^{-3.912} Z(u) = 0.02 Z(u)$$

## Shops Data

```
x_coord <- c(112, 823, 888, 105, 487)
y_coord <- c(198, 112, 846, 855, 537)
Chain <- rep("Chain 1", 5)
```

```
ID <- 1:length(x_coord)
df_shops <- data.frame(
  ID = ID,
  x_coord = x_coord,
  y_coord = y_coord,
  Chain = Chain,
  Gitter_ID = character(length(x_coord))
)
```

```
shops <- df_shops
ppp_shops <- ppp(x = shops$x_coord, y = shops$y_coord, window = window, marks = as.factor(shops$Chain))
ppp_shops <- rescale(ppp_shops, 1000, "km")
```

## Alternative Model

```
for (chain in chains_to_investigate) {
  #for (chain in levels(ppp_shops$marks)){
  print(chain)
  # Alternative Model
  ppp_chosen <- subset(ppp_shops, marks == chain, drop = TRUE)
  ppp_chosen <- unmark(ppp_chosen)

  # plot
  X <- layered(im_population, unmark(subset(ppp_shops, marks != chain, drop = TRUE)), ppp_chosen, ppp_o
  layerplotargs(X)[[1]] <- list(col = brewer.pal(n = 8, name = "Greys"), breaks = c(0, 1, 10, 50, 100, 1
  layerplotargs(X)[[2]] <- list(pch = 18, cex = 0.8, col = "#386f9c")
  layerplotargs(X)[[3]] <- list(pch = 18, cex = 1.5, col = "gold")
  layerplotargs(X)[[4]] <- list(pch = 20, col = "red2", cex = 1.5)
  plot(X, main = "Potential sources and cases", axes = TRUE, xlim = c(0, 1), ylim = c(0, 1) )

  ls_all_raisins = list()
  for (i in 1:ppp_chosen$n) {
    ls_all_raisins[i] = paste0("log((1 + abs(alpha) * (exp(-(abs(beta)) * ((x- ", ppp_chosen$x[i], ")^2
  }

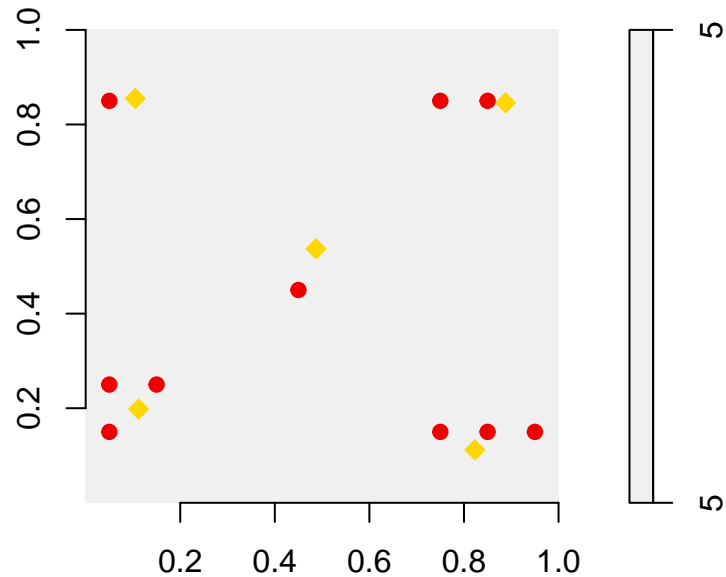
  str_all_raisins <- paste(ls_all_raisins, collapse = "+")
  eval(parse(text = paste('raisin_func <- function(x, y, alpha, beta) {(' , str_all_raisins , ')}', sep

  fit1 <- ippm(Q ~ offset(log(im_population) + raisin_func),
    start = list(alpha = 1, beta = 1), nlm.args = list(stepmax = 1), gcontrol = glm.control(
  print(paste("Alternative Model for ", chain))
  print(fit1)
  print(paste("Anova for ", chain))
}
```

```
print(anova(fit0, fit1, test = "LRT"))
}
```

```
## [1] "Chain 1"
```

## Potential sources and cases



```
## [1] "Alternative Model for Chain 1"
## Nonstationary Poisson process
## Fitted to point pattern dataset 'Q'
##
## Log intensity: ~offset(log(im_population) + raisin_func)
##
## Fitted trend coefficient: (Intercept) = 0.6560832
##
## Irregular parameters (covfunargs) fitted by 'ippm':
## alpha = -0.007440344
## beta = -3.473749e-07
##      Estimate      S.E.    CI95.lo CI95.hi Ztest    Zval
## (Intercept) 0.6560832 0.3162278 0.03628815 1.275878    * 2.074717
## [1] "Anova for Chain 1"
## Analysis of Deviance Table
##
## Model 1: ~offset(log(im_population))      Poisson
## Model 2: ~offset(log(im_population) + raisin_func)  Poisson
##   Npar Df    Deviance Pr(>Chi)
## 1     1
```

## 2 3 2 -1.6384e-08