Libraries

```r
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'
```

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

Definition of input Data

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

```r
scenario <- 1
```

Read Data

Window

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

Outbreak

Outbreak made with Diffusion Model:

```
#outbreak_data <- pd$read_pickle(paste("C:\\Users\\srude\\Documents\\Dev Kram\\Toy Example Traceback Mo
# outbreak_data <- pd$read_pickle("./Data/Outbreaks/Toy_Outbreak_10.pkl")
# ppp_outbreak <- ppp(x=outbreak_data$x_centroid, y=outbreak_data$y_centroid, window = window)
```

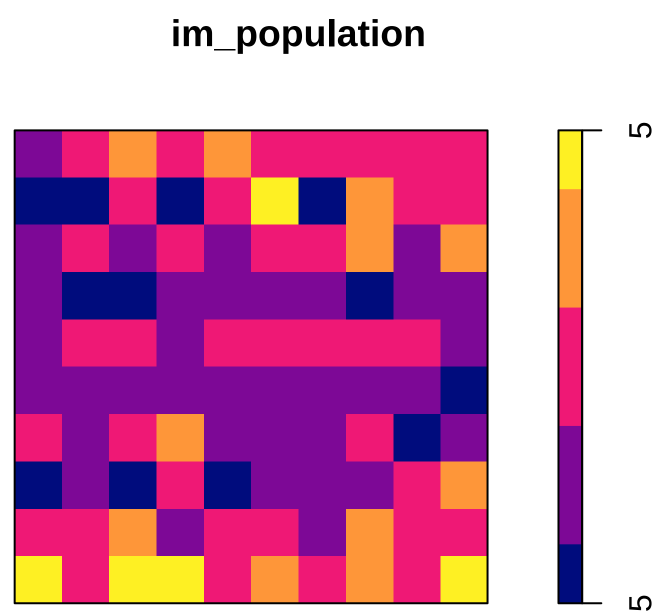
Outbreak artificially made to test model:

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

Population Data

```
im_population <- readRDS(paste("./Data/Population_Data/im_population_", as.character(scenario), ".rds",
im_population <- eval.im(im_population / 100)
im_population <- eval.im(pmax(im_population, 1e-10))
```

```
plot(im_population)
```



Quadrature Scheme

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

Null Model

```r
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

Private Computer

```r
shops <- pd$read_pickle(paste("C:\\Users\\srude\\Documents\\Pattern Comparison Project\\Toy_Example\\Di
```

Work Computer

```r
shops <- pd$read_pickle(paste("C:\\Users\\Sandra.Rudeloff\\Documents\\Pattern Comparison Project\\Toy_E
```

```r
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

```r
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)
```

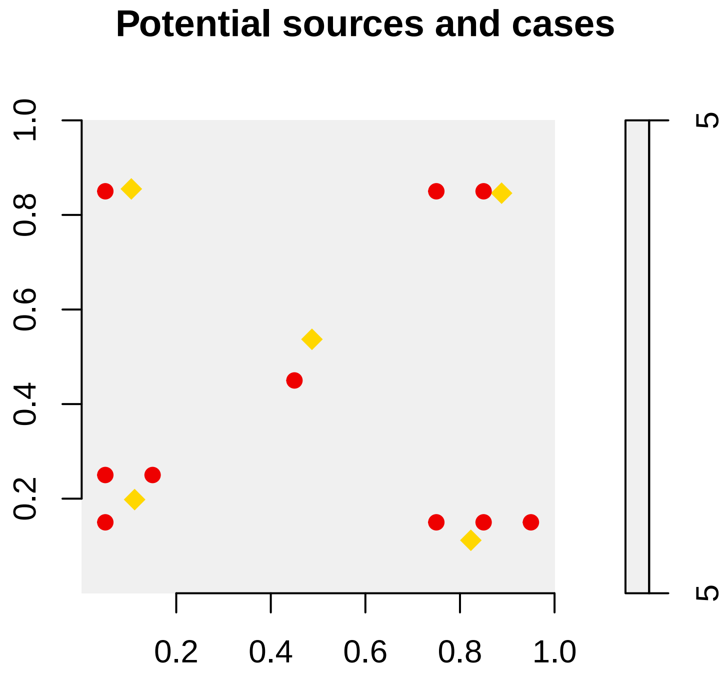
```
  # plot
    X <- layered(im_population, unmark(subset(ppp_shops, marks != chain, drop = TRUE)),ppp_chosen,ppp_o
  layerplotargs(X)[[1]] <- list(col = brewer.pal(n = 8, name = "Greys"), breaks = c(0, 1, 10, 50, 100, 
  layerplotargs(X)[[2]] <- list(pch = 18, cex = 0.8, col = "#386f9c")
  layerplotargs(X)[[3]] <- 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 = 5, beta = 2), 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"
```



```
## [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) = -4.541447
##
## Irregular parameters (covfunargs) fitted by 'ippm':
## alpha = 1.84886
## beta = -8.124726e-07
##              Estimate      S.E.   CI95.lo   CI95.hi Ztest      Zval
## (Intercept) -4.541447 0.3162278 -5.161242 -3.921652   *** -14.36132
## [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 -3.3673e-06
```