Raised Incidence Model for Hamburg Case

This script implements the raised incidence model by Diggle (1990) and the extensions to accommodate multiple potential sources by Diggle & Rowlingson (1994) on Data of the region of Hamburg.

- Population Data as the population at risk $[\lambda_0(x)]$

- Retailer location data as sources $[x_k]$

- Outbreak case locations as outbreak $[\lambda_1(x)]$

All spatial data is stored in the coordinate system ETRS89-LAEA Europe - EPSG:3035.

1 Load Libraries

```
library(spatstat)
library(sf)
```

2 Load Outbreak from Shapefile

```
sf_outbreak <- st_read("./Data/Outbreak/Aldi_Outbreak.shp")
```

```
## Reading layer 'Aldi_Outbreak' from data source
##   'C:\Users\srude\Documents\Dev Kram\Traceback_Model\Data\Outbreak\Aldi_Outbreak.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 20 features and 1 field
## Geometry type: POINT
## Dimension:     XY
## Bounding box:  xmin: 4312050 ymin: 3372150 xmax: 4331650 ymax: 3394450
## Projected CRS: ETRS89-extended / LAEA Europe
```
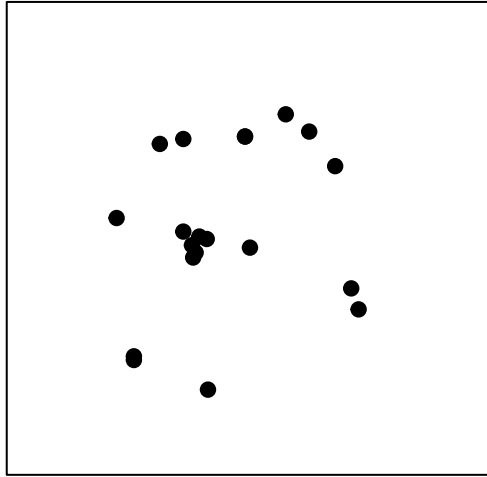
Convert into spatstat ppp object

```
ppp_outbreak <- as.ppp(X=sf_outbreak$geometry, W=owin(c(4303150,4342650), c(3365250,3403550)))
ppp_outbreak
```

```
## Planar point pattern: 20 points
## window: rectangle = [4303150, 4342650] x [3365250, 3403550] units
```

Plot

```
plot(ppp_outbreak, pch = 19,  main = "Artificial Outbreak")
```

Artificial Outbreak
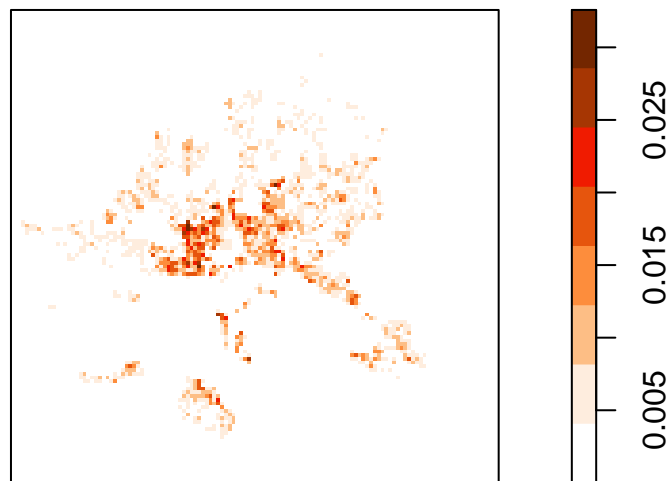

3 Load Population Data

We applied kernel smoothing (Diggle, 1985) to the population data to obtain an unnormalised estimate of the spatially varying population density of susceptibles ($\lambda_0(x)$).

```
im_population <- readRDS("./Data/im_population.rds")
```
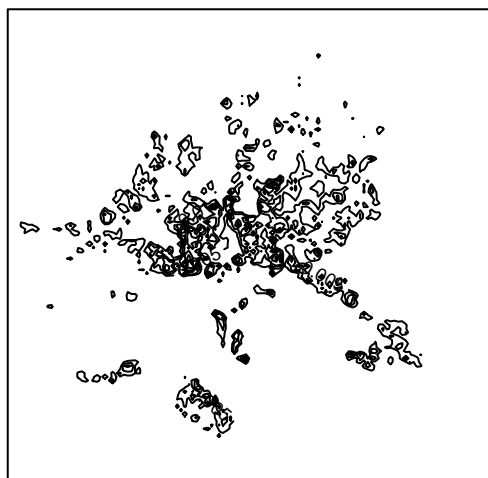
Plot

```
colors <- c("#ffffff", "#FEEDDE", "#FDBE85", "#FD8D3C", "#E6550D","#F01B00", "#A63603", "#732600")
plot(im_population, main = "Kernel Smoothed Population Density Hamburg 2011", col =colors)
```

Kernel Smoothed Population Density Hamburg 2011


```
contour(im_population, main = "Contour of Population Density Hamburg 2011")
```

Contour of Population Density Hamburg 2011

4   Load Potential Pattern

For now, we only use the one point source model, so we read only the Shapefile of one store of the chain.

```
sf_stores <- st_read("./Data/Potential Pattern/Aldi_store_6.shp")
```

```
## Reading layer 'Aldi_store_6' from data source
##   'C:\Users\srude\Documents\Dev Kram\Traceback_Model\Data\Potential Pattern\Aldi_store_6.shp'
##   using driver 'ESRI Shapefile'
## Simple feature collection with 1 feature and 33 fields
## Geometry type: POINT
## Dimension:      XY
## Bounding box:  xmin: 4316914 ymin: 3385953 xmax: 4316914 ymax: 3385953
## Projected CRS: ETRS89-extended / LAEA Europe
```
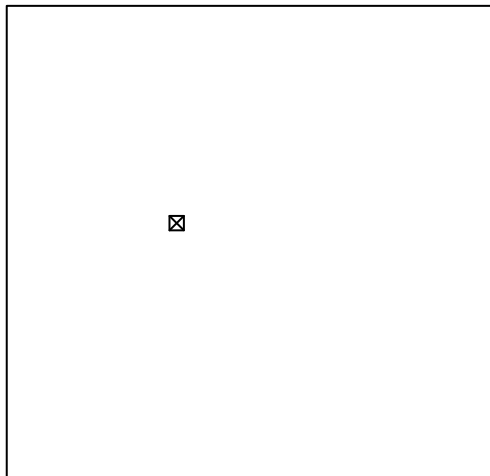
Convert into spatstat ppp object

```
ppp_stores <- as.ppp(X=sf_stores$geometry, W=owin(c(4303150,4342650), c(3365250,3403550)))
ppp_stores
```

```
## Planar point pattern: 1 point
## window: rectangle = [4303150, 4342650] x [3365250, 3403550] units
```

Plot

```
plot(ppp_stores, pch = 7, main = "Source")
```

Source


```
print(ppp_stores$x[1])
```

```
## [1] 4316914
```

```
print(ppp_stores$y[1])
```

```
## [1] 3385953
```

5   Fit Null Model

$H_0 : \lambda_1(x) = \rho\lambda_0(x)$

```
fit0 <- ppm(ppp_outbreak ~ offset(log(im_population)))
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in glm(fmla, family = quasi(link = "log", variance = "mu"), weights =
## .mpl.W, : fitting to calculate the null deviance did not converge -- increase
## 'maxit'?
```

```
fit0
```

```
## Nonstationary Poisson process
##
## Log intensity:  ~offset(log(im_population))
##
## Fitted trend coefficient:  (Intercept) = 320.0382
##
##               Estimate          S.E.   CI95.lo   CI95.hi Ztest        Zval
## (Intercept) 320.0382 2.294984e-73 320.0382 320.0382    *** 1.394511e+75
## *** Fitting algorithm for 'glm' did not converge ***
```

6   Fit Alternative Model

$H_0 : \lambda_1(x) = \rho\lambda_0(x)(1 + \alpha\ e^{-\beta x' x})$

We define the squared distance to the source:

```
d2source <- function(x, y, xPOS=ppp_stores$x[1], yPOS = ppp_stores$y[1]) {
  (x - xPOS)^2 + (y - yPOS)^2}
```

We define the functional form of the raised incidence part:

```
raisin <- function(x, y, alpha, beta) {
  1 + alpha * exp(-beta * d2source(x, y))
}
```

We fit the model to our outbreak

```
fit1 <- ippm(ppp_outbreak ~ offset(log(im_population) + log(raisin)),
  start = list(alpha = 5, beta = 1))
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning in glm(fmla, family = quasi(link = "log", variance = "mu"), weights =
## .mpl.W, : fitting to calculate the null deviance did not converge -- increase
## 'maxit'?

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning in glm(fmla, family = quasi(link = "log", variance = "mu"), weights =
## .mpl.W, : fitting to calculate the null deviance did not converge -- increase
## 'maxit'?

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning in glm(fmla, family = quasi(link = "log", variance = "mu"), weights =
## .mpl.W, : fitting to calculate the null deviance did not converge -- increase
## 'maxit'?

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning in glm(fmla, family = quasi(link = "log", variance = "mu"), weights =
## .mpl.W, : fitting to calculate the null deviance did not converge -- increase
## 'maxit'?

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning in glm(fmla, family = quasi(link = "log", variance = "mu"), weights =
## .mpl.W, : fitting to calculate the null deviance did not converge -- increase
## 'maxit'?
```

```
fit1
```

```
## Nonstationary Poisson process
##
## Log intensity:  ~offset(log(im_population) + log(raisin))
##
## Fitted trend coefficient:  (Intercept) = 320.0382
##
## Irregular parameters (covfunargs) fitted by 'ippm':
## alpha = 5
## beta = 1
##             Estimate         S.E.  CI95.lo  CI95.hi Ztest        Zval
## (Intercept) 320.0382 2.294984e-73 320.0382 320.0382   *** 1.394511e+75
## *** Fitting algorithm for 'glm' did not converge ***
```

6.1 Statistical Inference comparing the two models

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

```
## Analysis of Deviance Table
##
## Model 1: ~offset(log(im_population))     Poisson
## Model 2: ~offset(log(im_population) + log(raisin))    Poisson
##   Npar Df Deviance Pr(>Chi)
## 1    1
## 2    3  2         0        1
```