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 was initially stored in the coordinate system ETRS89-LAEA Europe - EPSG:3035 and transformed from metres to kilometres.

1 Load Libraries

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

2 Load Outbreak from Shapefile

```
sf_outbreak <- st_read("./Data/Outbreaks/Aldi_20/Aldi_Outbreak.shp")
```

```
## Reading layer 'Aldi_Outbreak' from data source
##   'C:\Users\srude\Documents\Dev Kram\Traceback_Model\Data\Outbreaks\Aldi_20\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

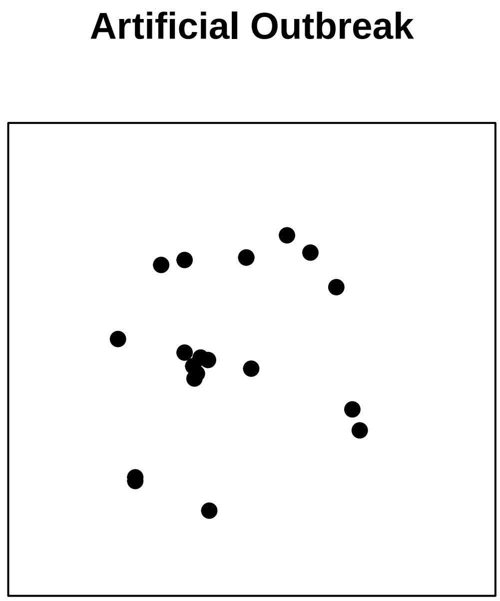
```
ppp_outbreak <- as.ppp(X=sf_outbreak$geometry, W=owin(c(4303150,4342650), c(3365250,3403550)))
ppp_outbreak
```

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

```
ppp_outbreak <- rescale(ppp_outbreak, 1000, "km")
```

Plot

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



3 Load Population Data

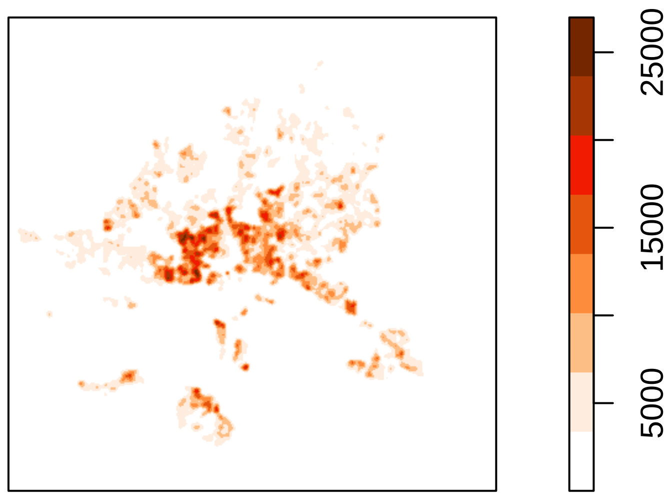
We applied kernel smoothing (Diggle, 1985) to the population data to obtain an unnormalised estimate of the spatially varying population density of susceptibles ($\lambda_0(x)$).

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

Plot

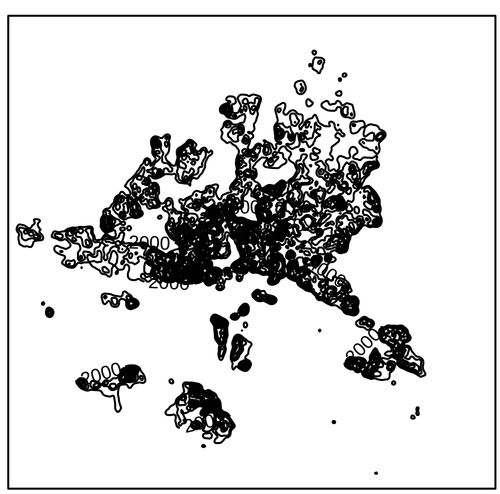
```
colors <- c("#ffffff", "#FEEDDE", "#FDBE85", "#FD8D3C", "#E6550D","#F01B00", "#A63603", "#732600")
plot(im_population, main = "Kernel Smoothed Population Density Hamburg 2011", col =colors)
```

Kernel Smoothed Population Density Hamburg 2011




```r
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

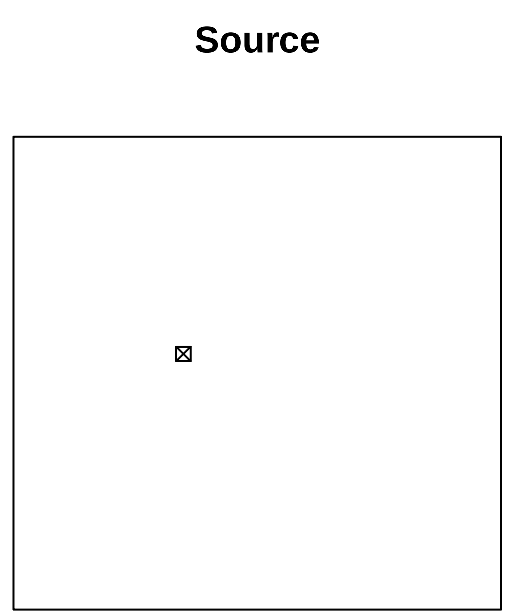
```
ppp_stores <- as.ppp(X=sf_stores$geometry, W=owin(c(4303150,4342650), c(3365250,3403550)))
ppp_stores
```

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

```
ppp_stores <- rescale(ppp_stores, 1000, "km")
```

Plot

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



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

```
## [1] 4316.914
```

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

```
## [1] 3385.953
```

5 Fit Null Model

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

```
fit0 <- ppm(ppp_outbreak ~ offset(log(im_population)), gcontrol = glm.control(maxit = 10000))
```

```
fit0
```

```
## Nonstationary Poisson process
##
## Log intensity:  ~offset(log(im_population))
##
## Fitted trend coefficient:  (Intercept) = -11.41112
##
##               Estimate      S.E.   CI95.lo   CI95.hi Ztest      Zval
## (Intercept) -11.41112 0.2236068 -11.84938 -10.97286   *** -51.03207
```

6 Fit Alternative Model

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

We define the squared distance to the source:

```
d2source <- function(x, y, xPOS= 4316.914, yPOS=3385.953) {
  (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), gcontrol = glm.control(maxit = 1000))
```

```
fit1
```

```
## Nonstationary Poisson process
##
## Log intensity:  ~offset(log(im_population) + log(raisin))
##
## Fitted trend coefficient:  (Intercept) = -11.5194
##
## Irregular parameters (covfunargs) fitted by 'ippm':
## alpha = 1.301572
## beta = 0.1413264
##              Estimate      S.E.   CI95.lo   CI95.hi Ztest     Zval
## (Intercept) -11.5194 0.2236068 -11.95766 -11.08113   *** -51.5163
```

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  0.79405   0.6723
```

7 Quadscheme

```
Q <- quadscheme(ppp_outbreak, eps = 100)
```

```
fit0_test <- ppm(Q ~ offset(log(im_population)), gcontrol = glm.control(maxit = 10000))
fit0_test
```

```
## Nonstationary Poisson process
##
## Log intensity:  ~offset(log(im_population))
##
## Fitted trend coefficient:  (Intercept) = -11.41112
##
##               Estimate      S.E.   CI95.lo   CI95.hi Ztest      Zval
## (Intercept) -11.41112 0.2236068 -11.84938 -10.97286   *** -51.03207
```

```
fit1_test <- ippm(Q ~ offset(log(im_population) + log(raisin)),
  start = list(alpha = 5, beta = 1), gcontrol = glm.control(maxit = 1000))
fit1_test
```

```
## Nonstationary Poisson process
##
## Log intensity:  ~offset(log(im_population) + log(raisin))
##
## Fitted trend coefficient:  (Intercept) = -11.5194
##
## Irregular parameters (covfunargs) fitted by 'ippm':
## alpha = 1.301572
## beta = 0.1413264
##              Estimate      S.E.   CI95.lo   CI95.hi Ztest     Zval
## (Intercept) -11.5194 0.2236068 -11.95766 -11.08113   *** -51.5163
```

8 Multiple Sources

```
all_raisins = list()
for (i in 1:3){#length(ppp_stores)  {
    all_raisins[i] =  paste0("(1 + alpha * exp(-beta * ((x- ",ppp_stores$x[i],")^2 + (y- ",ppp_stores$y[i]
}

raisin_func <- function(x, y, alpha, beta) {
  paste(all_raisins, collapse = "*")
}
```

```
d2source <- function(x, y, xPOS, yPOS) {
  (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))
}
```