Noname manuscript No.
(will be inserted by the editor)

Non-parametric spatial intensity estimation with R

Jorge Mateu · Mehdi Moradi ·

Received: date / Accepted: date

Abstract This paper

Keywords Bandwidth · Curve estimation · Kernel · Linear netwrok · Replicated point patterns · Time series · Trajectories · Voronoi ·

1 Introduction

The rest of the paper is organised as follows. In Section 2, we go through the details of different kernel- and Voronoi-based techniques for the purpose of intensity estimation when analysing spatial point patterns on $\mathbb{R}^2$. Section 3 is devoted to intensity estimation for point patterns on linear networks. In Section 4, we discuss different families of point patterns, and consider intensity estimation for replicated point patterns, time series of point patterns, and trajectory patterns. The paper ends with a discussion in Section 5

2 Spatial point patterns on $\mathbb{R}^2$

Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}, n < \infty$, be an observed realization of any point process on $\mathbb{R}^2$ within the window $W \subset \mathbb{R}^2$.

2.1 Kernel-based intensity estimators

The two most frequently used kernel-based intensity estimators for spatial point patterns on $\mathbb{R}^2$ are

$$\widehat{\lambda}(u)^{\mathrm{U}} = \frac{1}{c_W(u)} \sum_{i=1}^{n} \kappa(u - x_i), \quad u \in W, \tag{1}$$

and

$$\widehat{\lambda}(u)^{\mathrm{JD}} = \sum_{i=1}^{n} \frac{\kappa(u - x_i)}{c_W(x_i)}, \quad u \in W, \tag{2}$$

Grants or other notes

Jorge Mateu
Department of Mathematics, University Jaume I, Castellón, Spain.
E-mail: mateu@uji.es

Mehdi Moradi
Department of Statistics, Computer Science and Mathematics,
Institute of Advanced Materials and Mathematics (InaMat^2), Public University of Navarre, Pamplona, Spain.
E-mail: mehdi.moradi@unavarra.es

where κ is a probability density on $\mathbb{R}^2$, and

$$c_W(u) = \int_W \kappa(u - v)\mathrm{d}v, \quad u \in W, \tag{3}$$

plays the role of an edge corrector.

2.2 Marked point patterns (schoenberg)

2.3 Bandwidth selection

We next discuss a few of the most considered bandwidth selection approaches.

2.3.1 Scott's rule

2.3.2 Likelihood cross validation

2.3.3 Diggle's approach

2.3.4 Cronie and van Lieshout's criterion

2.4 Example 1: Wildfire in Nepal

```
library(sf)
library(maptools)
library(spatstat)
library(raster)
```

```
setwd("~/Desktop/Mehdi/Chapter book/Data/Fire/Data/Boundary")
w_nepal <- read_sf("NationalBoundary.shp")
w_nepal <- st_transform(w_nepal, crs = 3857)
winn_nepal <- as.owin(as_Spatial(w_nepal))


setwd("~/Desktop/Mehdi/Chapter book/Data/Fire/Data/ForestFire")
fp_nepal <- read_sf("ForestFire_Nepal.shp")
fp_nepal <- st_transform(fp_nepal, crs = 3857)
year_nepal <- fp_nepal$Year
```

```
xy_nepal <- st_coordinates(fp_nepal)
X_nepal <- ppp(x=xy_nepal[, 1], y=xy_nepal[, 2], marks = year_nepal, window=winn_nepal)
X_nepal <- as.ppp(X_nepal)

X_nepal <- X_nepal[X_nepal$marks==2016]
X_nepal <- unmark(X_nepal)

plot(X_nepal,pch=20,main="",lwd=4,cex=3)
```

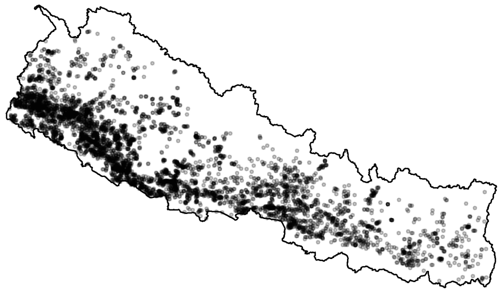

Fig. 1 Wildfire data in Nepal during 2016.

```
diggle_nepal <- bw.diggle(X_nepal)
ppl_nepal <- bw.ppl(X_nepal)
scott_nepal <- bw.scott(X_nepal)
CvL_nepal <- bw.CvL(X_nepal)
```

```
d_diggle_nepal <- density(X_nepal,  sigma = diggle_nepal,  positive=TRUE)
d_ppl_nepal <- density(X_nepal,  sigma = ppl_nepal,  positive=TRUE)
d_scott_nepal <- density(X_nepal,  sigma = scott_nepal,  positive=TRUE)
d_cvl_nepal <- density(X_nepal,  sigma = CvL_nepal,  positive=TRUE)
```

```
d_diggle_dig_nepal <- density(X_nepal,  sigma = diggle_nepal,  positive=TRUE,
                              diggle=TRUE)
d_ppl_dig_nepal <- density(X_nepal,  sigma = ppl_nepal,  positive=TRUE,
                           diggle=TRUE)
d_scott_dig_nepal <- density(X_nepal,  sigma = scott_nepal,  positive=TRUE,
                             diggle=TRUE)
d_cvl_dig_nepal <- density(X_nepal,  sigma = CvL_nepal, positive=TRUE,
                           diggle=TRUE)
```

```
d_diggle_epa_nepal <- density(X_nepal, sigma = diggle_nepal, positive=TRUE,
                              kernel="epanechnikov")
d_ppl_epa_nepal <- density(X_nepal, sigma = ppl_nepal, positive=TRUE,
                           kernel="epanechnikov")
d_scott_epa_nepal <- density(X_nepal, sigma = scott_nepal, positive=TRUE,
                             kernel="epanechnikov")
d_cvl_epa_nepal <- density(X_nepal, sigma = CvL_nepal, positive=TRUE,
                           kernel="epanechnikov")
```

```
d_diggle_dig_epa_nepal <- density(X_nepal, sigma = diggle_nepal, positive=TRUE,
                                  diggle=TRUE, kernel="epanechnikov")
d_ppl_dig_epa_nepal <- density(X_nepal, sigma = ppl_nepal, positive=TRUE, diggle=TRUE,
                               kernel="epanechnikov")
d_scott_dig_epa_nepal <- density(X_nepal, sigma = scott_nepal, positive=TRUE,
                                 diggle=TRUE, kernel="epanechnikov")
d_cvl_dig_epa_nepal <- density(X_nepal, sigma = CvL_nepal, positive=TRUE, diggle=TRUE,
                               kernel="epanechnikov")
```

```
sp_int_nepal <- stack(raster(d_diggle_nepal), raster(d_ppl_nepal), raster(d_scott_nepal),
                      raster(d_cvl_nepal), raster(d_diggle_dig_nepal),
                      raster(d_ppl_dig_nepal), raster(d_scott_dig_nepal),
                      raster(d_cvl_dig_nepal), raster(d_diggle_epa_nepal),
                      raster(d_ppl_epa_nepal), raster(d_scott_epa_nepal),
                      raster(d_cvl_epa_nepal), raster(d_diggle_dig_epa_nepal),
                      raster(d_ppl_dig_epa_nepal), raster(d_scott_dig_epa_nepal),
```

```
                        raster(d_cvl_dig_epa_nepal))
sp_int_nepal <- sp_int_nepal*10000000
names(sp_int_nepal) <- c("diggle_gaus","ppl_gaus","scott_gaus","CvL_gaus",
                         "diggle_gaus_JD", "ppl_gaus_JD","scott_gaus_JD",
                         "CvL_gaus_JD", "diggle_epa","ppl_epa", "scott_epa",
                         "CvL_epa", "diggle_epa_JD","ppl_epa_JD","scott_epa_JD",
                         "CvL_epa_JD")

at <- c(seq(0,1.4,0.2),2,4,7)
spplot(sp_int_nepal,  at=at,  colorkey=list(labels=list(cex=3)),
       col.regions = rev(topo.colors(20)),
       scales=list(draw=F),
       par.strip.text=list(cex=3))
```

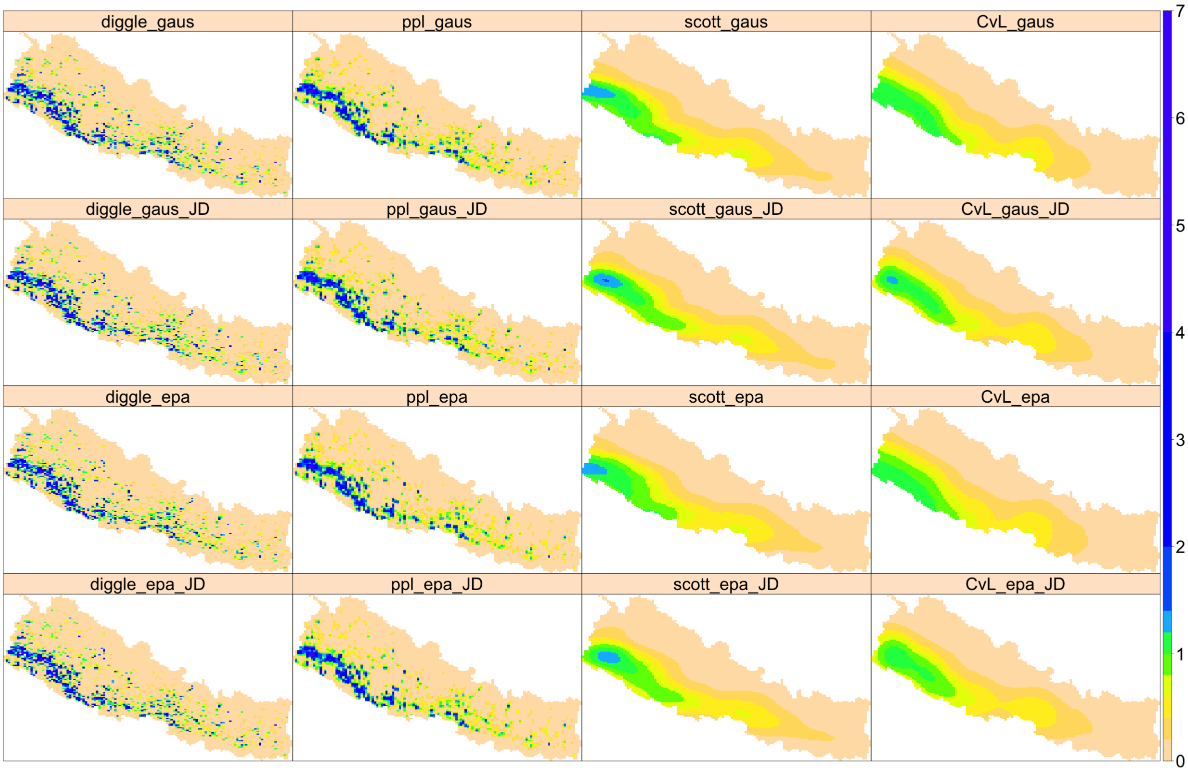

Fig. 2 Kernel-based intensity estimation for the wildfire data in Nepal during 2016. Layer's names start with bandwidth selection approach followed by the used kernel and edge-correction method.

2.5 Example 2: Crime in Medellín

```
cp_med <- read.csv("~/Desktop/Mehdi/Github/spatial-intensity-estimation/R/Medellin/Medellin.csv",
                   header = TRUE, sep =";",dec = ",")
year_med <- cp_med$Ano


winn_med <- read_sf("~/Desktop/Mehdi/Github/spatial-intensity-estimation/R/Medellin/BarrioVereda.shp
winn_med <- winn_med[winn_med$SUBTIPO_BA==1 & winn_med$LIMITECOMU!= "SN01",]
winn_med <- st_union(winn_med)
winn_med <- st_transform(winn_med,crs = 6257)
winn_med <- as.owin(as_Spatial(winn_med))
```

```r
year_med_id <- 2005
X_med <- ppp(cp_med$x[year_med==year_med_id],cp_med$y[year_med==year_med_id],
             window = winn_med)

plot(X_med,pch=20,main="",lwd=4,cex=3)
```

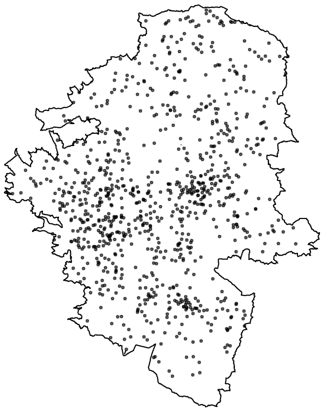

Fig. 3 Crime data in Medellín, Colombia, during 2005.

```r
diggle_med <- bw.diggle(X_med)
ppl_med <- bw.ppl(X_med)
scott_med <- bw.scott(X_med)
CvL_med <- bw.CvL(X_med)
```

```r
d_diggle_med <- density(X_med,  sigma = diggle_med,  positive=TRUE)
d_ppl_med <- density(X_med,  sigma = ppl_med,  positive=TRUE)
d_scott_med <- density(X_med,  sigma = scott_med,  positive=TRUE)
d_cvl_med <- density(X_med,  sigma = CvL_med,  positive=TRUE)
```

```r
d_diggle_dig_med <- density(X_med,  sigma = diggle_med,  positive=TRUE,
                               diggle=TRUE)
d_ppl_dig_med <- density(X_med,  sigma = ppl_med,  positive=TRUE,
                            diggle=TRUE)
d_scott_dig_med <- density(X_med,  sigma = scott_med,  positive=TRUE,
                              diggle=TRUE)
d_cvl_dig_med <- density(X_med,  sigma = CvL_med, positive=TRUE,
                            diggle=TRUE)
```

```r
d_diggle_epa_med <- density(X_med, sigma = diggle_med, positive=TRUE,
                               kernel="epanechnikov")
d_ppl_epa_med <- density(X_med, sigma = ppl_med, positive=TRUE,
                            kernel="epanechnikov")
d_scott_epa_med <- density(X_med, sigma = scott_med, positive=TRUE,
                              kernel="epanechnikov")
d_cvl_epa_med <- density(X_med, sigma = CvL_med, positive=TRUE,
                            kernel="epanechnikov")
```

```r
d_diggle_dig_epa_med <- density(X_med, sigma = diggle_med, positive=TRUE,
                                   diggle=TRUE, kernel="epanechnikov")
d_ppl_dig_epa_med <- density(X_med, sigma = ppl_med, positive=TRUE, diggle=TRUE,
                          kernel="epanechnikov")
d_scott_dig_epa_med <- density(X_med, sigma = scott_med, positive=TRUE,
```

```
                                        diggle=TRUE, kernel="epanechnikov")
d_cvl_dig_epa_med <- density(X_med, sigma = CvL_med, positive=TRUE, diggle=TRUE,
                           kernel="epanechnikov")
```

```
sp_int_med <- stack(raster(d_diggle_med), raster(d_ppl_med), raster(d_scott_med),
                      raster(d_cvl_med), raster(d_diggle_dig_med),
                      raster(d_ppl_dig_med), raster(d_scott_dig_med),
                      raster(d_cvl_dig_med), raster(d_diggle_epa_med),
                      raster(d_ppl_epa_med), raster(d_scott_epa_med),
                      raster(d_cvl_epa_med), raster(d_diggle_dig_epa_med),
                      raster(d_ppl_dig_epa_med), raster(d_scott_dig_epa_med),
                      raster(d_cvl_dig_epa_med))
sp_int_med <- sp_int_med*100000
names(sp_int_med) <- names(sp_int_nepal)

at <- c(0,0.6,0.8,1,1.2,1.4,1.6,1.8,2,2.5,3,6)
spplot(all, at=at, colorkey=list(labels=list(cex=3)),
       col.regions = rev(topo.colors(20)),
       scales=list(draw=F),
       par.strip.text=list(cex=3))
```

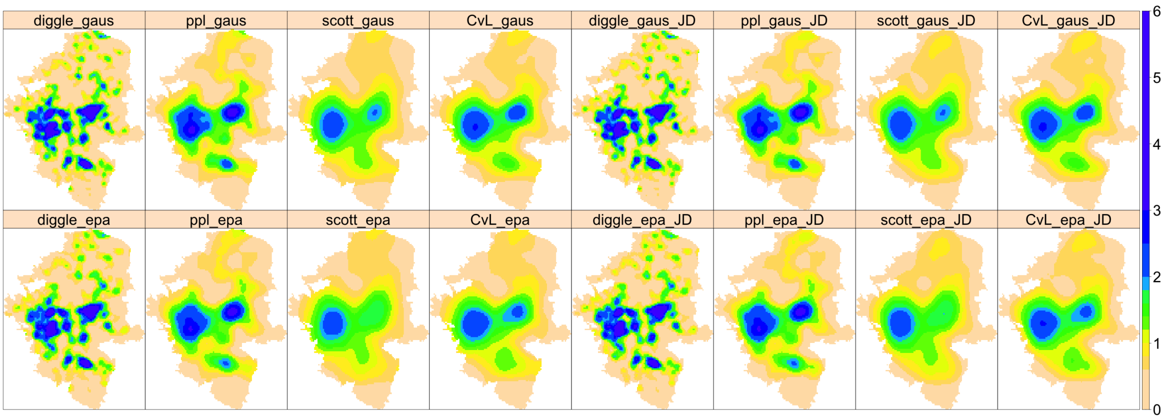

Fig. 4 Kernel-based intensity estimation for the crime data in Medellín, Colombia, during 2005. Layer's names start with bandwidth selection approach followed by the used kernel and edge-correction method.

2.6 Voronoi-based intensity estimators

```
d_vor_1_nepal <- densityVoronoi(X_nepal, f=1, nrep = 1)
d_vor_2_nepal <- densityVoronoi(X_nepal, f=0.8, nrep = 400)
d_vor_3_nepal <- densityVoronoi(X_nepal, f=0.6, nrep = 400)
d_vor_4_nepal <- densityVoronoi(X_nepal, f=0.5, nrep = 400)
d_vor_5_nepal <- densityVoronoi(X_nepal, f=0.4, nrep = 400)
d_vor_6_nepal <- densityVoronoi(X_nepal, f=0.2, nrep = 400)
d_vor_7_nepal <- densityVoronoi(X_nepal, f=0.1, nrep = 400)
d_vor_8_nepal <- densityVoronoi(X_nepal, f=0.05, nrep = 400)
```

```
sp_int_nepal_v <- stack(raster(d_vor_1_nepal), raster(d_vor_2_nepal),
                        raster(d_vor_3_nepal), raster(d_vor_4_nepal),
                        raster(d_vor_5_nepal), raster(d_vor_6_nepal),
                        raster(d_vor_7_nepal), raster(d_vor_8_nepal))
names(sp_int_nepal_v) <- NULL
```

```
names <-  as.character(sort(c(seq(.2,1,.2),0.1,0.05,0.5),decreasing = TRUE))
names <- paste("p =", names)
```

```
sp_int_nepal_v <- sp_int_nepal_v*10000000
at <- c(0, 0.3, 0.7,seq(2,5,1),30)
spplot(all_v, at=at, colorkey=list(labels=list(cex=3)),
       col.regions = topo.colors(20),
       scales=list(draw=F),
       par.strip.text=list(cex=3),
       names.attr=names)
```

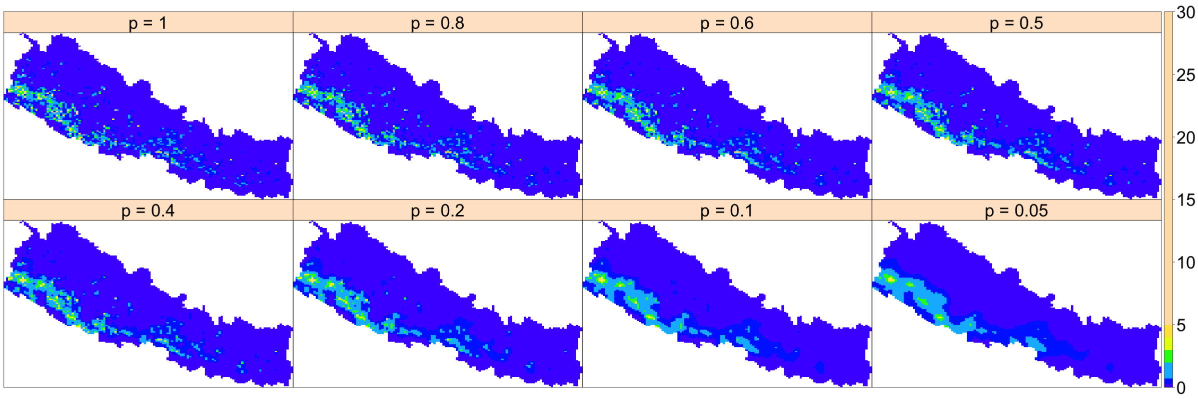

Fig. 5 Smoothed-Voronoi intensity estimation for the wildfire data in Nepal during 2016 based on different retention probability.

```
d_vor_1_med <- densityVoronoi(X_med, f=1, nrep = 1)
d_vor_2_med <- densityVoronoi(X_med, f=0.8, nrep = 400)
d_vor_3_med <- densityVoronoi(X_med, f=0.6, nrep = 400)
d_vor_4_med <- densityVoronoi(X_med, f=0.5, nrep = 400)
d_vor_5_med <- densityVoronoi(X_med, f=0.4, nrep = 400)
d_vor_6_med <- densityVoronoi(X_med, f=0.2, nrep = 400)
d_vor_7_med <- densityVoronoi(X_med, f=0.1, nrep = 400)
d_vor_8_med <- densityVoronoi(X_med, f=0.05, nrep = 400)
```

```
sp_int_med_v <- stack(raster(d_vor_1_med), raster(d_vor_2_med),
                      raster(d_vor_3_med), raster(d_vor_4_med),
                      raster(d_vor_5_med), raster(d_vor_6_med),
                      raster(d_vor_7_med), raster(d_vor_8_med))
names(sp_int_med_v) <- NULL
names <-  as.character(sort(c(seq(.2,1,.2),0.1,0.05,0.5),decreasing = TRUE))
names <- paste("p =", names)
```

```
sp_int_med_v <- sp_int_med_v*100000
at <- c(0,seq(1,3,0.5),seq(5,9,2),35)
spplot(all_v, at=at, colorkey=list(labels=list(cex=3)),
       col.regions = rev(topo.colors(20)),
       scales=list(draw=F),
       par.strip.text=list(cex=3),
       names.attr=names)
```

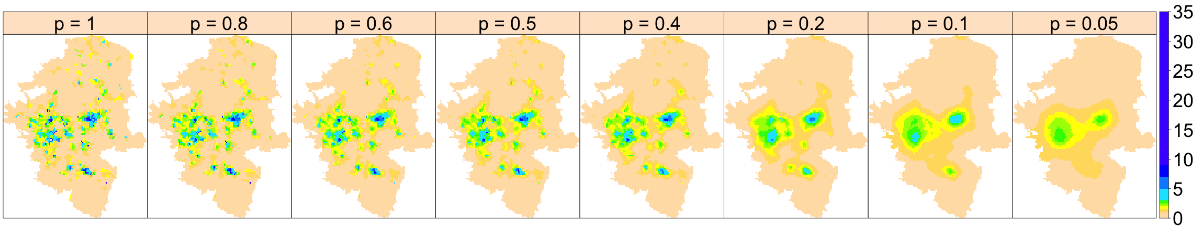

Fig. 6 Smoothed-Voronoi intensity estimation for the crime data in Medellín, Colombia, during 2005 based on different retention probability.

3 Spatial point patterns on linear networks

```
library(rgdal)

setwd("~/Desktop/Mehdi/Github/spatial-intensity-estimation/R/Valencia/shape files")
w_vlc <- readOGR("valencia_outline.shp")
w_vlc <- st_as_sf(w_vlc)
w_vlc <- st_transform(w_vlc, crs = 3857)
w_vlc <- st_sf(geom=w_vlc)
winn_vlc <- as.owin(as_Spatial(w_vlc))

net_vlc <- read_sf("valencia_road.shp")
net_vlc <- st_transform(net_vlc, crs = 3857)
net_vlc <- as.linnet.SpatialLines(as_Spatial(net_vlc))
Window(net_vlc) <- winn_vlc
net_vlc <- as.linnet(net_vlc, W=winn_vlc)

setwd("~/Desktop/Mehdi/Github/spatial-intensity-estimation/R/Valencia")

cp_vlc <- read.csv("obs_data.csv",header = TRUE, sep =",",dec = ",")
cp_vlc_year <- cp_vlc$year
cp_vlc_points <- SpatialPointsDataFrame(
                   coords = data.frame(x=as.numeric(cp_vlc$crime_lon),
                                       y=as.numeric(cp_vlc$crime_lat)),
                                     data = data.frame(rep(0,nrow(cp_vlc)))
                   )
proj4string(cp_vlc_points) <-  CRS("+proj=longlat")
cp_vlc_points <- st_as_sfc(cp_vlc_points)
cp_vlc_points <- st_as_sf(cp_vlc_points)
cp_vlc_points <- st_transform(cp_vlc_points, crs = 3857)
```

```
X_valencia <- lpp(st_coordinates(cp_vlc_points)[cp_vlc_year==2020,],L=net_vlc)
X_valencia <- unique(X_valencia)

plot(w_vlc$geometry,lwd=4)
plot(X_valencia,pch=20,main="",lwd=4,cex=1,add=T,cols="red",col="blue")
```

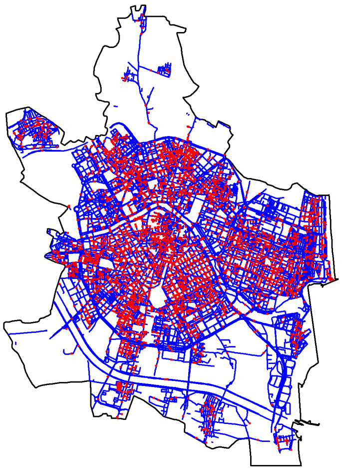

Fig. 7 Street crimes in Valencia, Spain, during 2020.

```
scott_valencia <- bw.scott.iso(X_valencia)
ppl_valencia <- bw.lppl(X_valencia)

d_ppl_valencia <- densityQuick.lpp(X_valencia,  sigma = ppl_valencia,
                                   positive=TRUE,dimyx=512)
d_scott_valencia <- densityQuick.lpp(X_valencia,  sigma = scott_valencia,
                                     positive=TRUE,dimyx=512)
```

```
par(mfrow=c(1,2))
plot(d_ppl_valencia,zlim=c(0,6),main="",ribwid=0.02,ribsep=0.005,
     leg.args=list(cex.axis=2.2),ribside=c("bottom"))
plot(w_vlc,add=T,lwd=4)
plot(d_scott_valencia,zlim=c(0,6),main="",ribwid=0.02,ribsep=0.005,
     leg.args=list(cex.axis=2.2),ribside=c("bottom"))
plot(w_vlc,add=T,lwd=4)
```

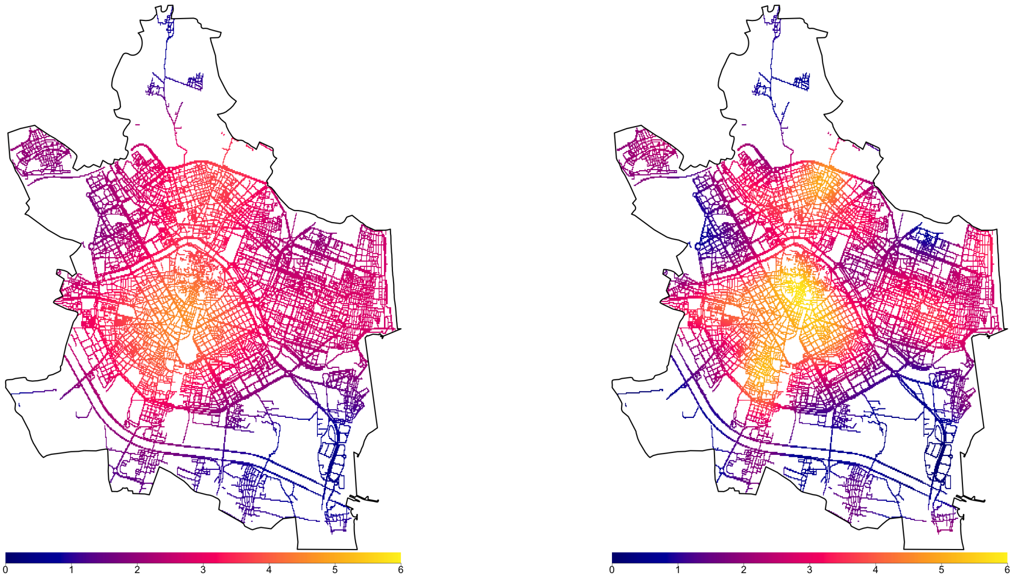

Fig. 8 Kernel-based intensity estimation for the crime data in Valencia, Spain, during 2020. Left: likelihood cross validation bandwidth selection, Right: Scott's rule for bandwidth selection.

4 Families of point patterns

4.1 Replicated point patterns

```
load("~/Desktop/Mehdi/Github/spatial-intensity-estimation/R/Bubbles/completedata.RData")
```

```
pps <- split(bubbles,f=list(as.factor(bubbles$VA),as.factor(bubbles$FC)))
```

```
library(ggplot2)
library(gridExtra)

window <- as.data.frame(pps$`5.5`$patterns$`1`$window)
plot_list <- list()

for (j in 1:9){

  df <- lapply(pps[[j]]$patterns, as.data.frame)
  df_row <- lapply(df, nrow)
  F1 <- cbind(df[[1]],FC=rep(pps[[j]]$FC[[1]],df_row[[1]]),
              VA=rep(pps[[j]]$VA[[1]],df_row[[1]]),
              npattern=rep(1,df_row[[1]]))

  for (i in 2:length(pps[[j]]$FC)) {

    F1 <- rbind(F1,
      cbind(df[[i]],FC=rep(pps[[j]]$FC[[i]],df_row[[i]]),
            VA=rep(pps[[j]]$VA[[i]],df_row[[i]]),
            npattern=rep(i,df_row[[i]]))
    )

  }

  plot_list[[j]] <- ggplot(F1, aes(x = x, y = y)) +
    geom_point(lwd=3) +
```

```
    geom_polygon(data = window, aes(x = x, y = y), alpha = 0.1)+
    facet_wrap(~ npattern, ncol = 2) +
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          strip.background = element_blank(),
          strip.text.x = element_blank(),
          legend.position = "none")
}
```

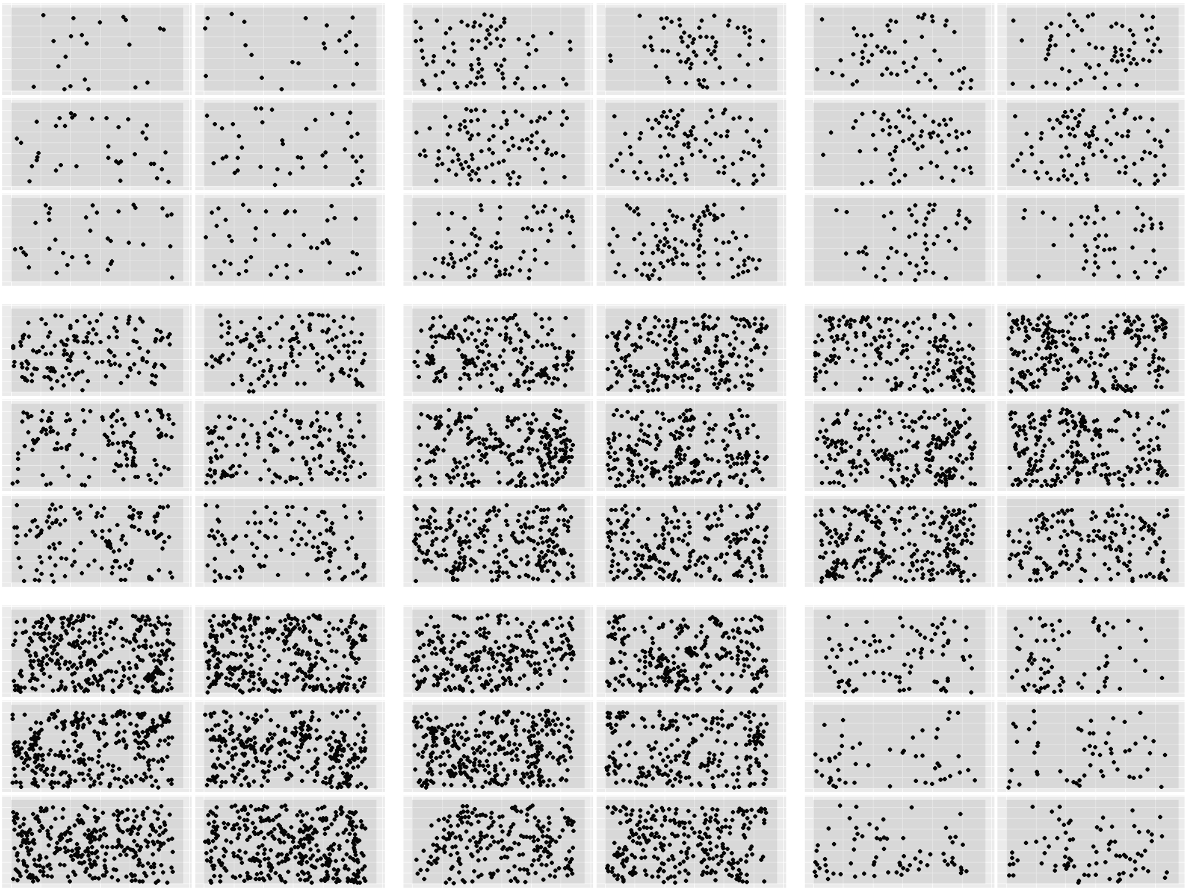

Fig. 9 Floating bubbles data.

```
sgs <- unlist(lapply(bubbles$patterns,bw.scott))
sgs_com <- exp(mean(log(sgs)))
```

```
im_lists <- list()
for (i in 1:9) {
  ps_list <- pps[[i]]$patterns
  im_list <- list()
  for (j in 1:6) {
    im_list[[j]] <- density(ps_list[[j]],sigma=sgs_com,positive=TRUE,diggle=TRUE)
  }
  im_lists[[i]] <- im_list
}

im_lists_means <- lapply(X=1:9,function(i){
```

```
  raster(Reduce("+",im_lists[[i]])/6)
})
```

```
im_lists_means_stack <- stack(im_lists_means)
names <- unique(paste(paste("VA =",as.factor(bubbles$VA)),
                       ",",paste("FC =", as.factor(bubbles$FC))))
spplot(im_lists_means_stack,col.regions=rev(terrain.colors(100)),
       colorkey=list(labels=list(cex=3)),
       scales=list(draw=F),
       par.strip.text=list(cex=3),
       names.attr=names)
```

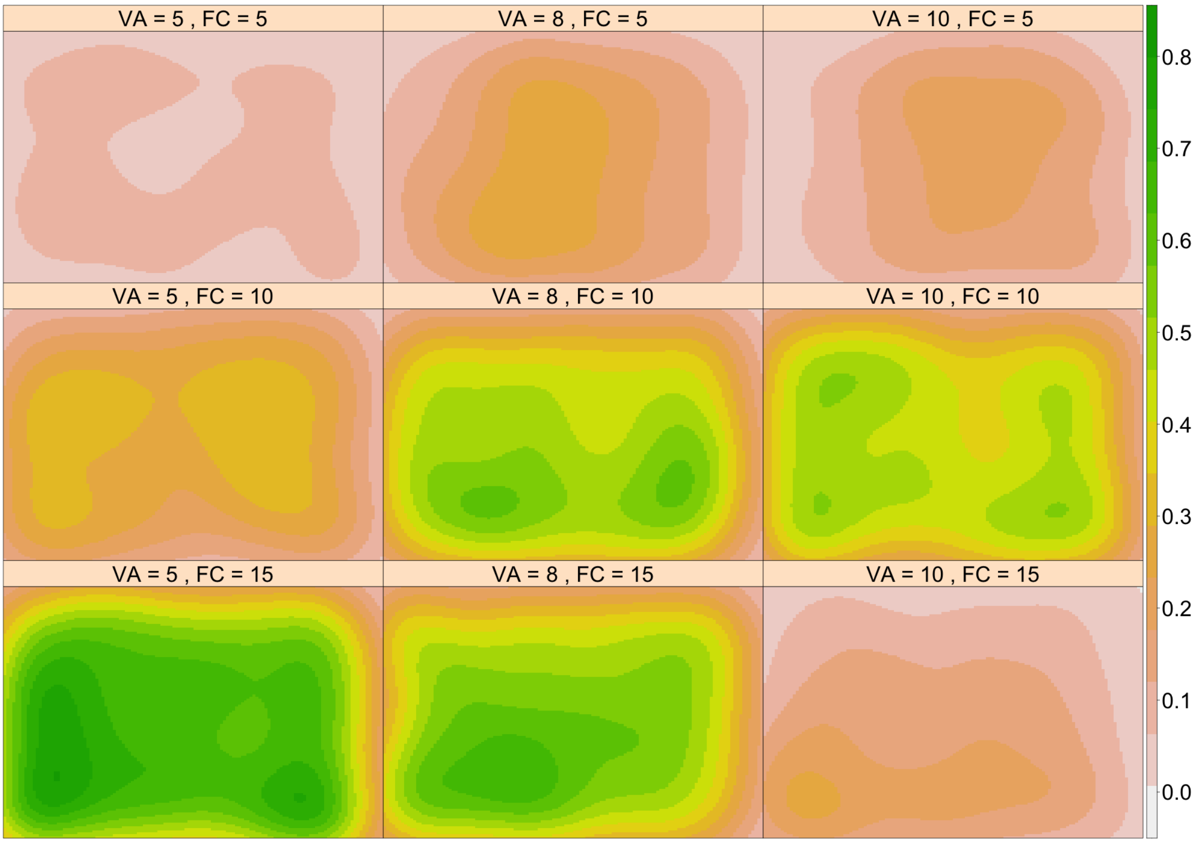

Fig. 10 Average estimated intensities for the floating bubbles data.

4.2 Time series of point patterns

```
setwd("~/Desktop/Mehdi/Github/spatial-intensity-estimation/R/Afghanistan")
X_Afghanistan <- read.csv("Dataset_AW_Month.csv",sep = ";")
X_Afghanistan <- X_Afghanistan[-which(is.na(X_Afghanistan[,1])),]
```

```
X_Afghanistan <- SpatialPointsDataFrame(X_Afghanistan[,1:2],
                                        data = data.frame(month=X_Afghanistan[,5],
                                                          year=X_Afghanistan[,6]))
proj4string(X_Afghanistan) = CRS("+proj=longlat")
```

```
X_Afghanistan <- st_as_sf(X_Afghanistan)
X_Afghanistan <- st_transform(X_Afghanistan,crs = 3857)
```

```
library(rnaturalearth)
winn_Afghanistan <- ne_countries(country = 'Afghanistan',returnclass = c("sf"))
winn_Afghanistan <- st_transform(winn_Afghanistan,crs = 3857)
X_Afghanistan <- st_intersection(X_Afghanistan,winn_Afghanistan)

winn_Afghanistan <- as.owin(as_Spatial(winn_Afghanistan))
```

```
xy <- st_coordinates(X_Afghanistan)
X_Afghanistan <- ppp(xy[,1],xy[,2],window = winn_Afghanistan,
                     marks = data.frame(month=X_Afghanistan$month,
                                        year=X_Afghanistan$year))
```

```
X_Afghanistan <- split(X_Afghanistan,f=as.factor(X_Afghanistan$marks$year))
X_Afghanistan <- lapply(X=1:6, function(i){
  split(X_Afghanistan[[i]],f=as.factor(X_Afghanistan[[i]]$marks$month))
})
X_Afghanistan <- unlist(X_Afghanistan, recursive=FALSE)

par(mfrow=c(6,12))
for (i in 1:length(X_Afghanistan)) {
  plot(unmark(X_Afghanistan[[i]]),pch=20,cex=3,main="",lwd=4)
}
```

Fig. 11 Afghan war data.

```
CvL_Afghanistan <- lapply(X=1:length(X_Afghanistan), function(i){
  bw.CvL(X_Afghanistan[[i]],ns=50)
})

CvL_Afghanistan <- unlist(CvL_Afghanistan)
CvL_Afghanistan_com <- exp(mean(log(CvL_Afghanistan)))
```

```
d_Afghanistan <- lapply(X=1:length(X_Afghanistan), function(i){
  d <- density(X_Afghanistan[[i]],sigma = CvL_Afghanistan_com,
                diggle = TRUE,positive = TRUE)
  raster(d)
})

d_Afghanistan <- stack(d_Afghanistan)
```

```
Afghan_cities <- readOGR("~/Desktop/Mehdi/Github/spatial-intensity-estimation/R/Afghanistan/provicin
proj4string(Afghan_cities) = CRS("+proj=longlat")
Afghan_cities <- st_as_sf(Afghan_cities)
Afghan_cities <- st_transform(Afghan_cities, crs = 3857)
```

```
KABUL <- Afghan_cities[Afghan_cities$PRV_NAME=="KABUL",]
KANDAHAR <- Afghan_cities[Afghan_cities$PRV_NAME=="KANDAHAR",]
GHAZNI <- Afghan_cities[Afghan_cities$PRV_NAME=="GHAZNI",]
HIRAT <- Afghan_cities[Afghan_cities$PRV_NAME=="HIRAT",]

selected_cities <- st_union(st_union(KABUL,HIRAT),st_union(GHAZNI,KANDAHAR))
```

```
d_Afghanistan <- d_Afghanistan*10^9

names <- expand.grid(c(1:12),c(2004:2009))
library(latticeExtra)
spplot(d_Afghanistan,col.regions=rev(terrain.colors(100)),
       colorkey=list(labels=list(cex=3)),
       scales=list(draw=F),
       par.strip.text=list(cex=3),
       names.attr=paste(names[,1],"/",names[,2])
)+layer(sp.polygons(as_Spatial(selected_cities),lwd=2))
```

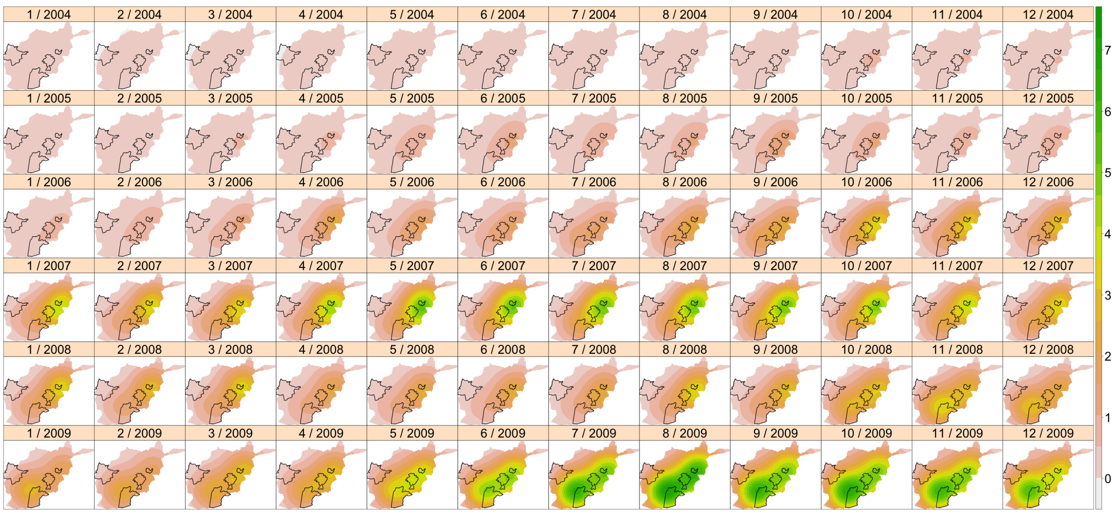

Fig. 12 Kernel-based intensity estimation for the Afghan war data.

```
KABUL_int <- extract(d_Afghanistan,as_Spatial(KABUL))[[1]]
GHAZNI_int <- extract(d_Afghanistan,as_Spatial(GHAZNI))[[1]]
KANDAHAR_int <- extract(d_Afghanistan,as_Spatial(KANDAHAR))[[1]]
HIRAT_int <- extract(d_Afghanistan,as_Spatial(HIRAT))[[1]]
```

```
par(mfrow=c(1,4))
matplot(t(KABUL_int),type = "l",ylab = "Intensity",xlab = "Time",cex.axis = 2,
        cex.lab=2,lwd=2,ylim=c(0,7.5))
matplot(t(GHAZNI_int),type = "l",ylab = "Intensity",xlab = "Time",cex.axis = 2,
        cex.lab=2,lwd=2,ylim=c(0,7.5))
matplot(t(KANDAHAR_int),type = "l",ylab = "Intensity",xlab = "Time",cex.axis = 2,
        cex.lab=2,lwd=2,ylim=c(0,7.5))
matplot(t(HIRAT_int),type = "l",ylab = "Intensity",xlab = "Time",cex.axis = 2,
        cex.lab=2,lwd=2,ylim=c(0,7.5))
```

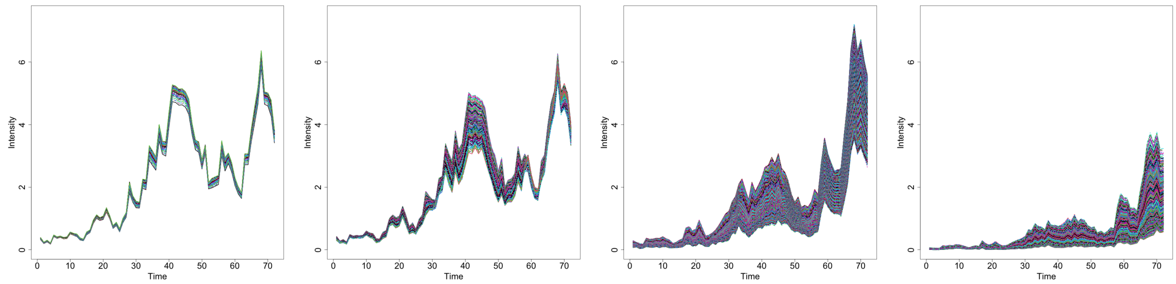

Fig. 13 Pixel time series of the estimated intensities. From left to right: Kabul, Ghazni, Kandahar, and Hirat

4.3 Trajectory patterns

```
library(trajectories)
library(spatstat)

load("D:/UPNA/Github/spatial-intensity-estimation/R/Trajectories/Beijing.small.RData")
```

```
pps <- as.Track.ppp(Beijing.small,timestamp = "20 mins")
bw <- lapply(pps, bw.scott)
```

```
sig <- exp(mean(log(unlist(bw))))
imlist <- lapply(pps, density.ppp, sigma=sig,positive=TRUE)
```

```
library(raster)
imlist_raster <- lapply(imlist, raster)
imlist_stack <- stack(imlist_raster)
imlist_stack <- imlist_stack*10^6
```

```
library(rgdal)
setwd("D:/UPNA/Github/spatial-intensity-estimation/R/Trajectories/shapes/china")
Bei <- readOGR("CHN_adm3.shp")
Beisp <- Bei[Bei$NAME_1=="Beijing",]
Beisp <- spTransform(Beisp, CRS("+proj=utm +zone=50 +ellps=WGS84 +datum=WGS84
+units=m +no_defs "))
library(chron)
names <- attr(pps,"tsq")
names <- times(format(names, "%H:%M:%S"))
```

```
library(latticeExtra)
names <- attr(pps,"tsq")
names <- times(format(names, "%H:%M:%S"))
spplot(imlist_stack,col.regions=rev(terrain.colors(100)),
       colorkey=list(labels=list(cex=3)),
```

```
       scales=list(draw=F),
       par.strip.text=list(cex=3),
       names.attr=names)+layer(sp.polygons(Beisp,lwd=2))
```

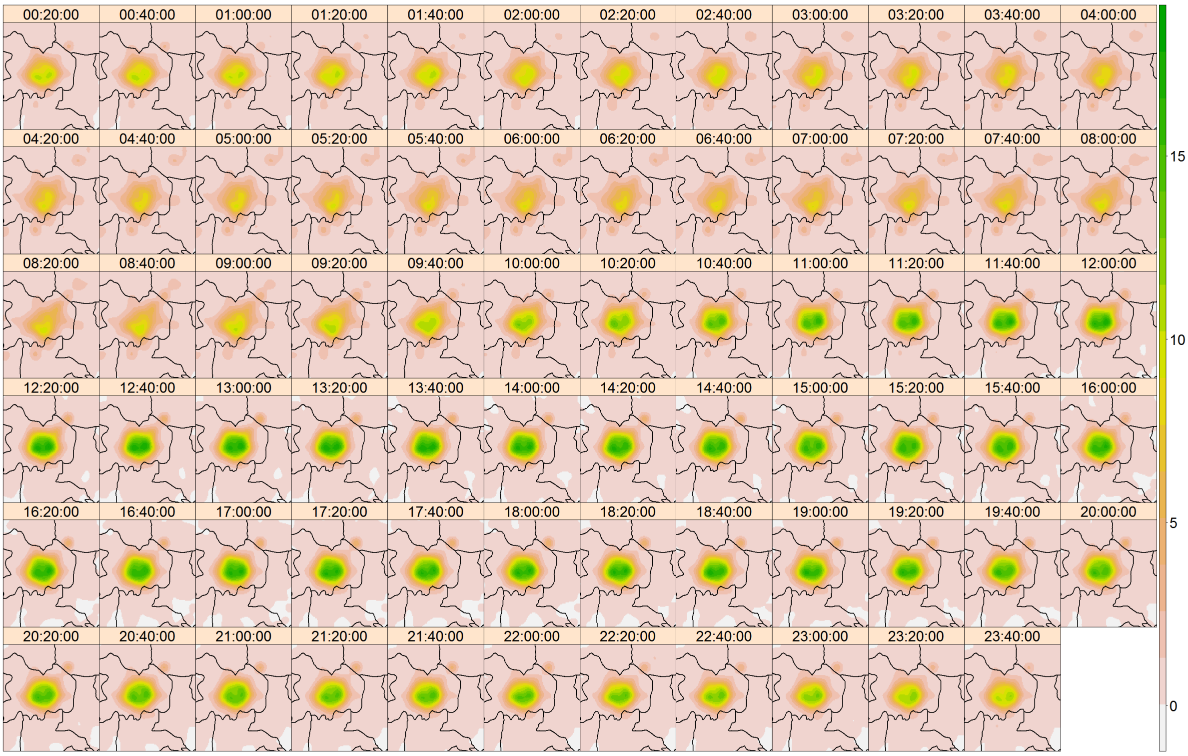

Fig. 14 Estimated intensities for the taxi data in Beijing, China during the 3rd of February 2008.

```
mean_int <- mean(imlist_stack)
```

```
setwd("D:/UPNA/Github/spatial-intensity-estimation/R/Trajectories/shapes/Beijing-shp/shape")

Bei_points <-  readOGR("points.shp")
Bei_points <- spTransform(Bei_points, CRS("+proj=utm +zone=50 +ellps=WGS84 +datum=WGS84
+units=m +no_defs "))
```

```
id <- c("taxi","attraction","school","viewpoint")
plot_list <- list()


for(j in 1:4){
   Bei_type <- Bei_points[Bei_points$type==id[j],]
   plot_list[[j]] <- spplot(mean_int,col.regions=rev(terrain.colors(100)),
                           colorkey=list(labels=list(cex=3)),
                           scales=list(draw=F),
                           par.strip.text=list(cex=3),
                           sp.layout = list(Bei_type,pch=20,cex = 3, col = "black"))
}
```

```
do.call("grid.arrange",c(plot_list,ncol=2))
```

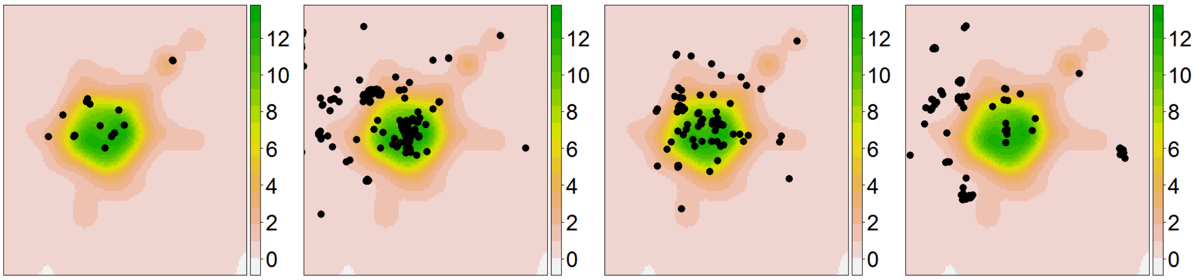

Fig. 15 Aderage estimated intensity for the taxi data in Beijing, China during the 3rd of February 2008. From left to right dark spots stand for points of taxi, attraction, school, and viewpoint.

5 Discussion