Topic 2: Spatio-temporal Point Processes

This topic introduces and investigates spatio-temporal point processes. In particular, we consider

- Introduction to **spatio-temporal point processes**.
- **R class ppx**.
- **Plotting** of spatio-temporal point process data.
- **Spatio-temporal Poisson process.**
- Using **north Cumbria data.**

Spatio-temporal point processes

Spatio-temporal point process data arise in many fields of application.

One general approach to analysing spatio-temporal point process data is to extend existing methods for purely spatial data by considering the time of occurrence as a distinguishing feature, or mark, attached to each event.

The events of a **spatio-temporal point process** form a set of points,

$$\mathcal{P} = \{f(s_i; t_i) : \ i = 1, 2, ..., \}$$

where $s_i \in \mathbb{R}^n$ is the spatial location and $t_i \in \mathbb{R}_+$ is the time of occurrence of the i-th event.

In practice, the data are the points $(x_i; t_i)$, $i = 1, ..., n$, that form the partial realization of the process restricted to a finite spatio-temporal domain of observation $S \times T$, where S is a spatial polygon and T is an interval.

Point patterns in multi-dimensional space-time

Support for multi-dimensional space-time point patterns has been added to `spatstat`. These points belong to the class `"ppx"` and are created by the function `ppx`. There may be any number of dimensions of space.

```
> library(spatstat)
> df <- data.frame(x = runif(100, max = 3), y = runif(100,
+ max = 3), z = runif(100,  max = 2), t = runif(100))
> bb <- boxx(c(0, 3), c(0, 3), c(0, 2), c(0, 1))
> X <- ppx(data=df,domain = bb,coord.type=c("s","s","s","t"))
> summary(X)
Multidimensional point pattern
100 points
3-dimensional space coordinates (x,y,z)
1-dimensional time coordinates (t)
Domain: 4-dimensional box:
[0, 3] x [0, 3] x [0, 2] x [0, 1] units
```

```
> marks(X) <- 1:100
> X
Multidimensional point pattern
100 points
3-dimensional space coordinates (x,y,z)
1-dimensional time coordinates (t)
1 column of marks: marks
Domain:
4-dimensional box:
[0, 3] x [0, 3] x [0, 2] x [0, 1] units
> head(X$data)
Hyperframe:
x         y         z           t marks
1 2.9184107 0.8831184 0.2313848 0.52816110     1
2 0.4989395 0.7300519 1.1675523 0.27023625     2
3 2.9729564 1.3597320 1.6384882 0.58517409     3
...
```

Plotting of spatio-temporal point process data

The most effective form of display for a spatio-temporal point process data is an animation, repeated viewing of which may yield insights that are not evident in static displays. Nevertheless, static displays are sometimes useful summaries.

The **stpp** package includes various display functions.

The data-set **fmd,** included in `stpp`, contains a three-column matrix of spatial locations and reported days (from 1 February 2001) of foot-and-mouth disease outbreaks in the county of Cumbria.

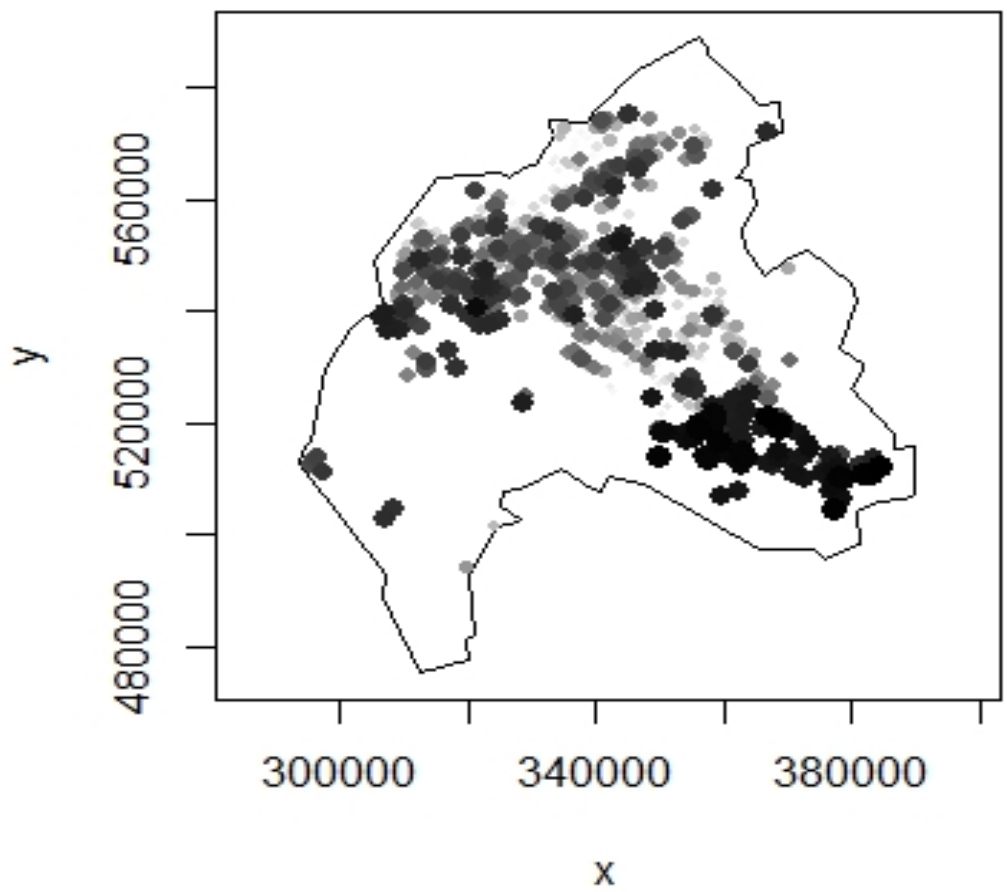
At the beginning of the epidemic the cumulative number of cases increases slowly, because the virus can be transmitted only over short distances and few of the susceptible farms are within range of the early cases.

This is followed by a period of rapid increase, as the infected area spreads and there are correspondingly more susceptible farms within the transmission range.

Finally, the rate of spread slows down as the epidemic is brought under control through a combination of reactive culling of infected animals and pre-emptive culling of animals at nearby farms.

Static display of the data can be obtained by the following R commands.

```
> library(stpp)
> data(fmd)
> str(fmd)
> data(northcumbria)
> fmd <- as.3dpoints(fmd)
> plot(fmd, s.region = northcumbria, pch=19, mark=TRUE)
```



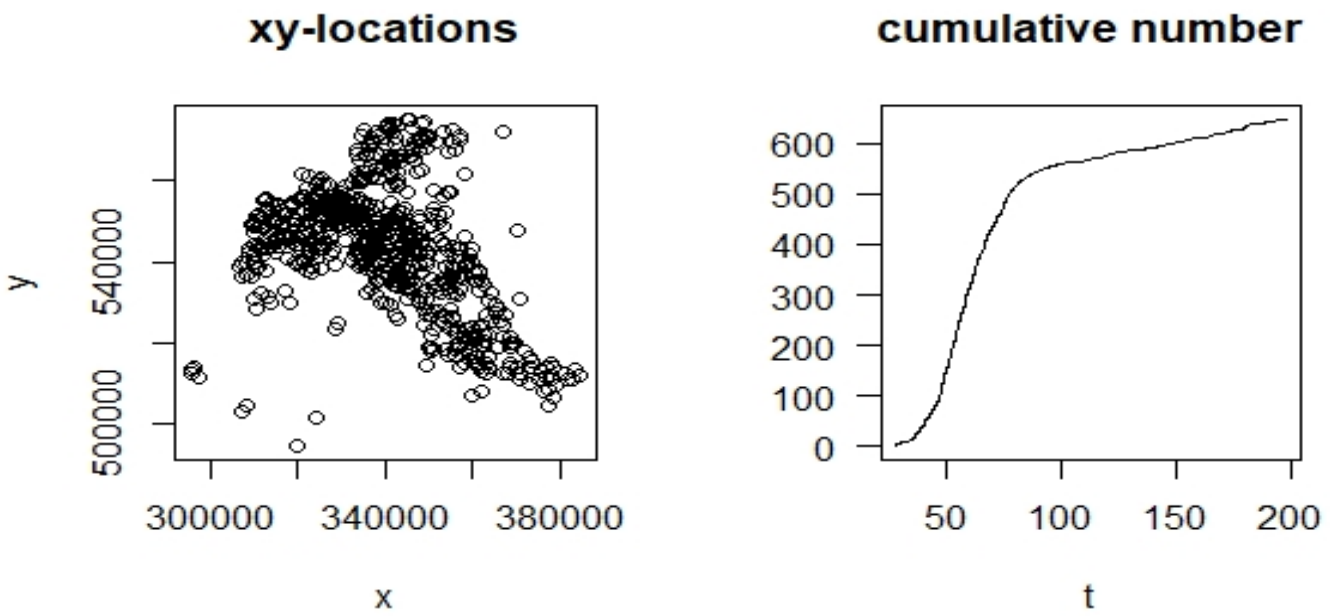
Time is treated as a quantitative mark: light grey dots correspond to the oldest events and dark grey ones correspond to the most recent events.

The function animation provides an animation of a space-time point pattern.

```
> animation(fmd,runtime=40,cex=0.5,s.region = northcumbria)
```

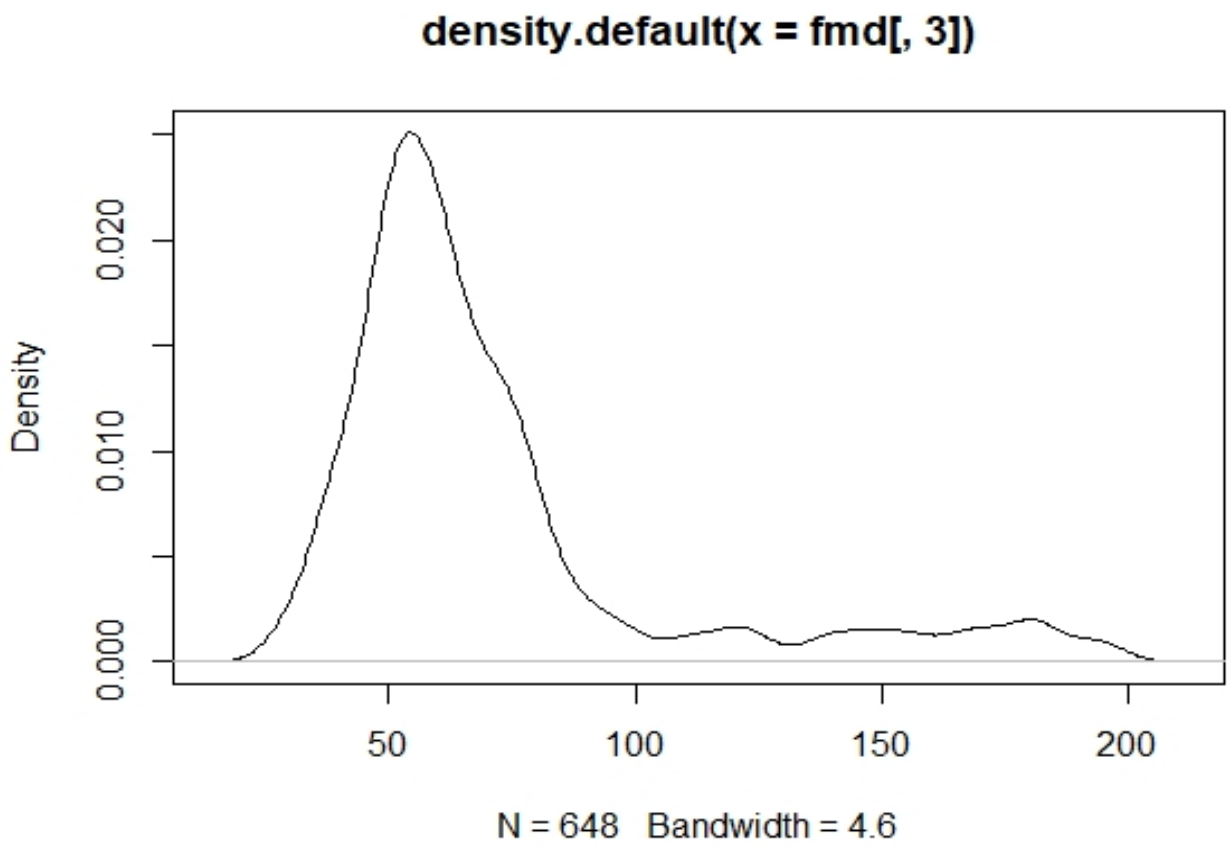
A two-panels plot showing spatial locations and cumulative times can be obtained using the command

```
> plot(fmd)
```



The temporal intensity function estimated from the fmd dataset shows the intensity of foot-and-mouth disease outbreaks in time.

```
> plot(density(fmd[,3]))
```



Spatio-temporal Poisson process

Spatio-temporal Poisson processes can be simulated by the function **rpp.** Realizations are simulated in a region $S \times T$, where S is a polygon and T is an interval, with default the unit cube.

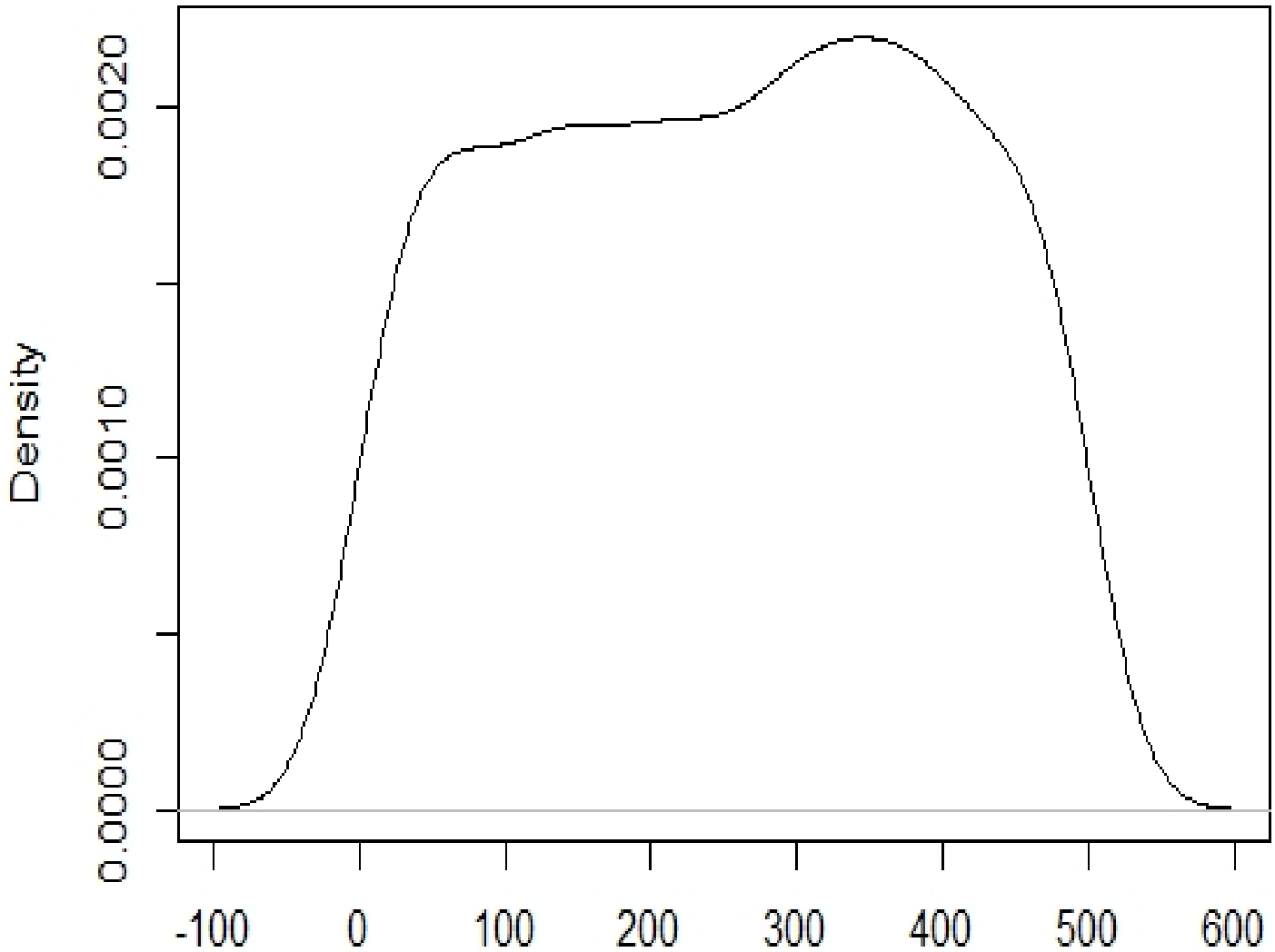
For a homogeneous Poisson process, the intensity is a constant. For example, the sequence of commands

```
> hpp2 <- rpp(npoints = 1000, s.region = northcumbria,
+ t.region = c(1, 500), discrete.time = TRUE)
> animation(hpp2$xyt, runtime=40, s.region = hpp2$s.region)
```

generates and displays a realization of the Poisson process with 1000 points in the region $S \times T$, where S is the county of Cumbria and $T = [1; 500]$. The argument `npoints` specifies the number of events to be generated.

The temporal intensity function can be obtain by

```
> plot(density(hpp2$xyt[,3]))
```



The function **rpp** can also generate realizations of inhomogeneous Poisson processes. The intensity is then specified by a function of the coordinates and times.
For example, the sequence of commands

```
> lbda1 <- function(x, y, t) {1000* exp(-4 * y)*exp(-2*t)}
> ipp1 <- rpp(lambda = lbda1)
> animation(ipp1$xyt,runtime = 40)
```

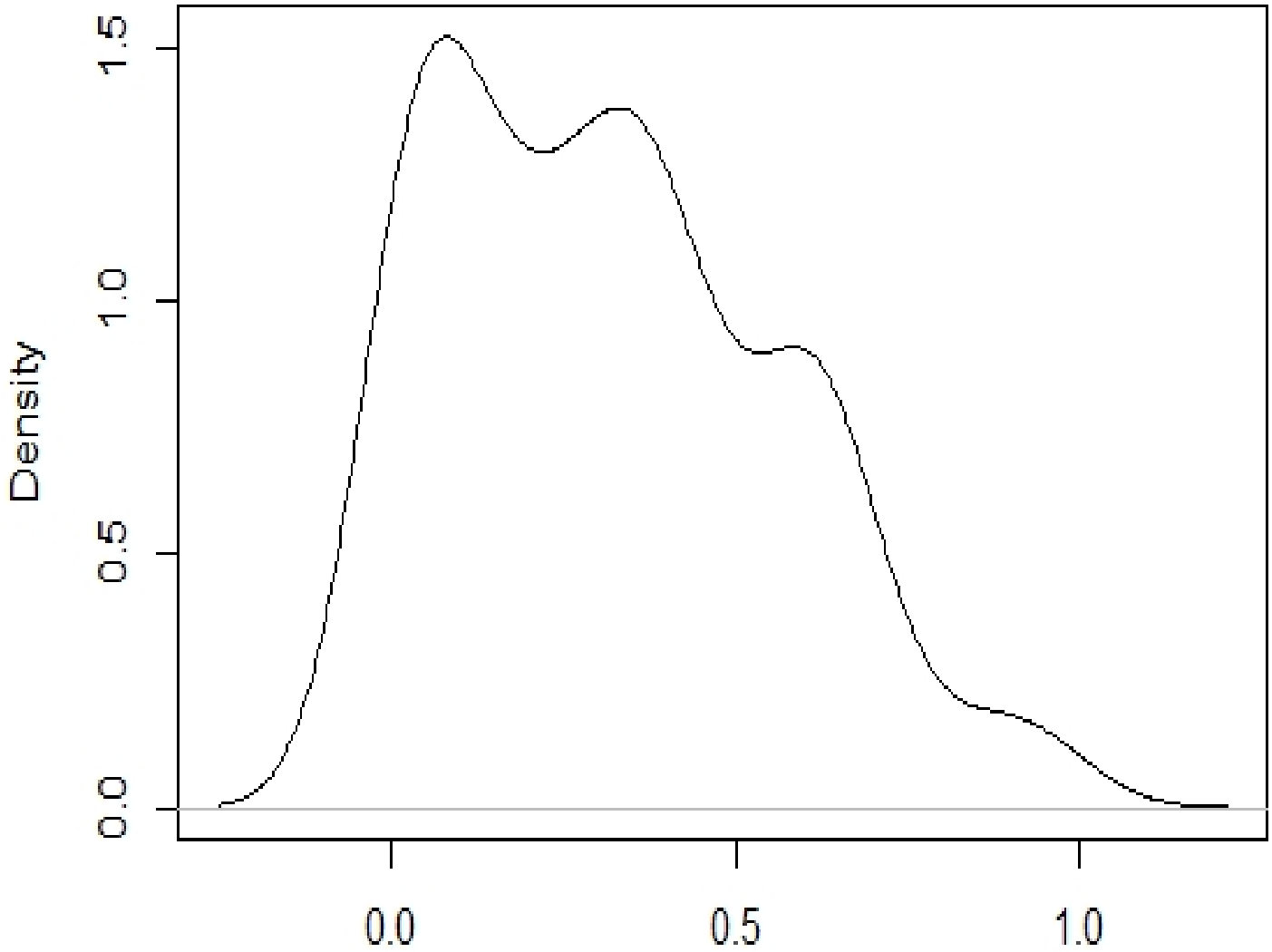
generates the Poisson process with intensity

$$\lambda(x, y, t) = 1000 \cdot exp(-4y) \cdot exp(-2t)$$

in the unit cube.

The temporal intensity functions can be obtain by the following R command

```
> plot(density(ipp1$xyt[,3]))
```


Density
0.0
0.5
1.0
1.5
0.0
0.5
1.0

Key R commands

boxx(...)	*creates a multi-dimensional box*
ppx(data,...)	*creates a multidimensional space-time point pattern*
fmd	*data set gives the spatial locations and reported times of food-and-mouth disease*
northcumbria	*data set gives the boundary of the county of north Cumbria (UK)*
as.3dpoints(...)	*creates data in spatio-temporal point format*
animation(xyt,...)	*provides animation of spatio-temporal point patterns*
rpp(lambda,...)	*generates Poisson point patterns*
density(...)	*computes kernel density estimates*