

STPP

Représentation, analyse et simulation de processus ponctuels spatio-temporels

Edith Gabriel

Université d'Avignon et des Pays de Vaucluse

En collaboration avec Peter J Diggle et Barry Rowlingson

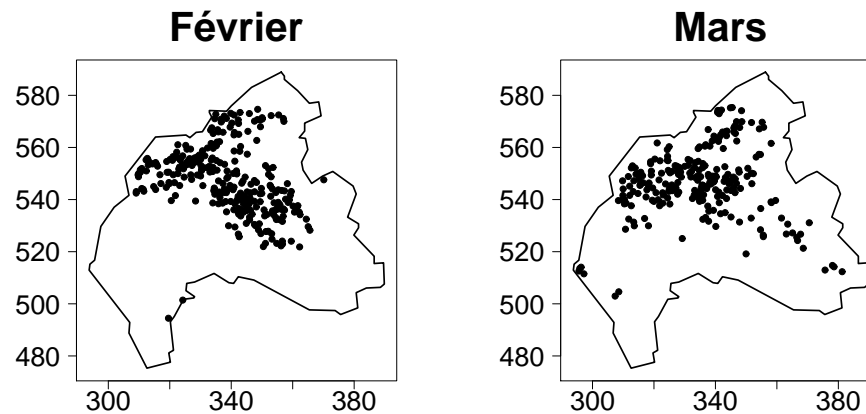
Lancaster University



Motivations

Données ponctuelles spatio-temporelles

- ★ Type de données $x = (s, t)$, avec s une localisation spatiale et t un temps.
- ★ Ex : Cas de fièvre aphteuse (nord-ouest de l'Angleterre) depuis 01/01/2001
 - s : localisation des fermes infectées
 - t : jour de détection des cas de fièvre aphteuse dans la ferme



Questions relatives à ces données

- ★ La répartition des points est-elle aléatoire ?
- ★ Si non, y a-t-il compétition/répulsion
ou coopération/attraction entre les points ?
- ★ Quel modèle associer aux données ?

Plan

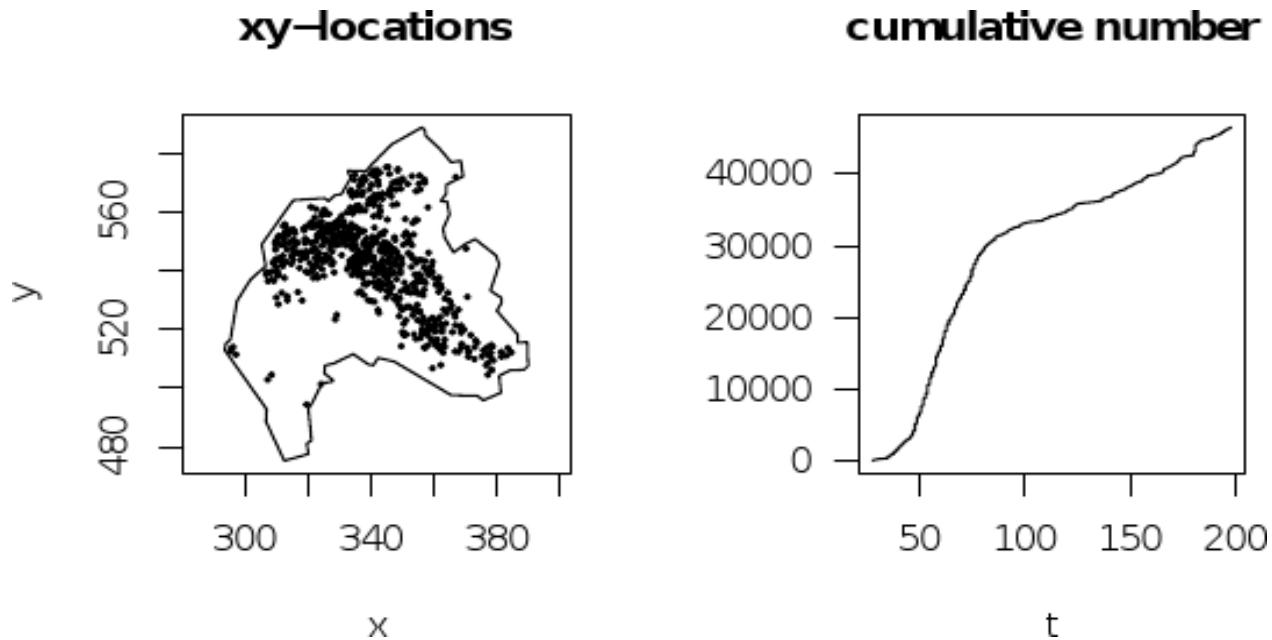
Le package `stpp` propose :

1. des outils de visualisation de données ponctuelles spatio-temporelles,
2. des modèles de processus ponctuels spatio-temporels,
3. un outil d'analyse de la structure de ces processus.

Représentation graphique des données

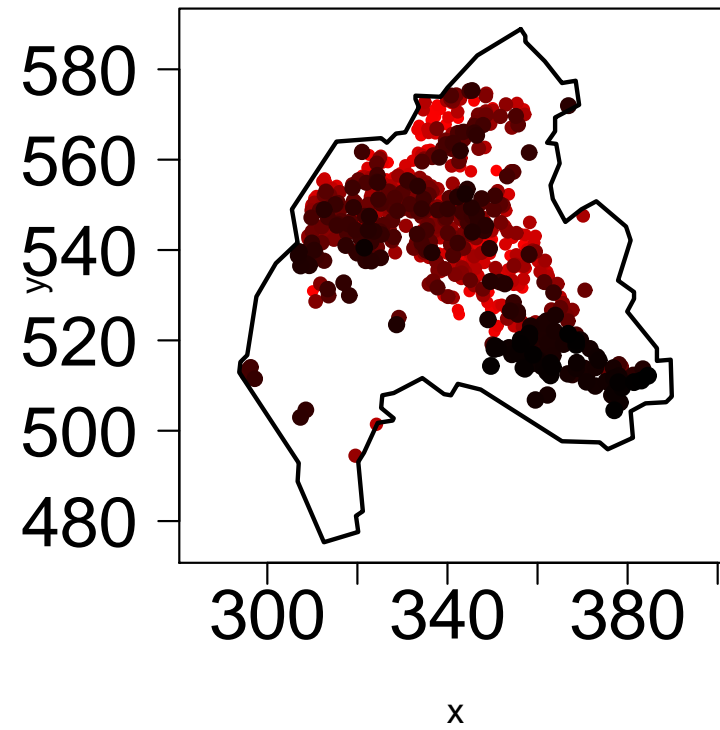
Représentation statique (1)

```
data("fmd"); data("northcumbria")  
fmd <- as.3dpoints(fmd)  
plot(fmd, s.region = northcumbria)
```



Représentation statique (2)

```
plot(fmd, mark = TRUE, mark.col = "red")
```



Représentations dynamiques

★ En 2D :

```
animation(fmd, runtime = 10, cex = 0.5, s.region = northcumbria)
```

★ En 3D :

```
library("rgl")
```

```
library("rpanel")
```

```
stan(fmd, twid = 0.01, bgpoly = northcumbria, bgframe = TRUE)
```

Modèles de processus ponctuels

Processus ponctuel spatio-temporel

★ Définition :

Mécanisme aléatoire générant un ensemble dénombrable de points de $\mathbb{R}^2 \times \mathbb{R}$.

En pratique, on n'en observe qu'une partie dans une fenêtre $\mathcal{D}_{ST} = S \times T$.

★ Notations :

- Φ : le processus ponctuel,
- x_1, \dots, x_n : une réalisation de n points de Φ , avec $x_i = (s_i, t_i)$
localisation spatiale $s_i \in S$, temps $t_i \in T$.
- $N(A) =$ nombre d'évènements dans A .

★ Intensité : $\lambda(x) = \lambda(s, t) = \lim_{|ds| \rightarrow 0, |dt| \rightarrow 0} \frac{\mathbb{E}[N(ds \times dt)]}{|ds||dt|}$,

$ds \times dt$: petit cylindre contenant x .

Processus de Poisson homogène

Il est utilisé comme référence de répartition complètement aléatoire.

★ Définition :

1. $N(\mathcal{D}_{ST})$ suit une loi de Poisson d'espérance $\lambda|\mathcal{D}_{ST}|$,
2. Sachant $N(\mathcal{D}_{ST}) = n$, les n points de \mathcal{D}_{ST} sont indépendants et issus de la loi uniforme sur \mathcal{D}_{ST} .

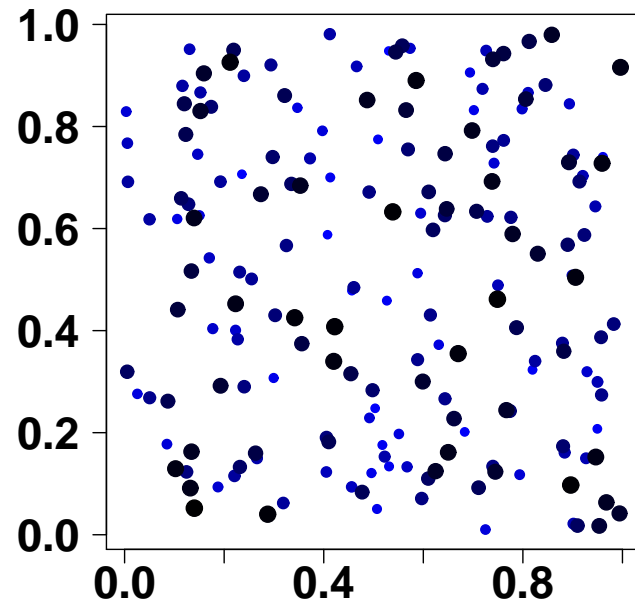
ou 2.' Si \mathcal{D}_{ST} et \mathcal{D}'_{ST} sont disjoints, $N(\mathcal{D}_{ST})$ et $N(\mathcal{D}'_{ST})$ sont indépendants.

★ Simulation :

1. Simuler le nombre d'évènements $n = N(\mathcal{D}_{ST})$ selon une distribution de Poisson d'espérance $\lambda|\mathcal{D}_{ST}|$.
2. Générer les n positions et temps selon la loi uniforme sur S et T respectivement.

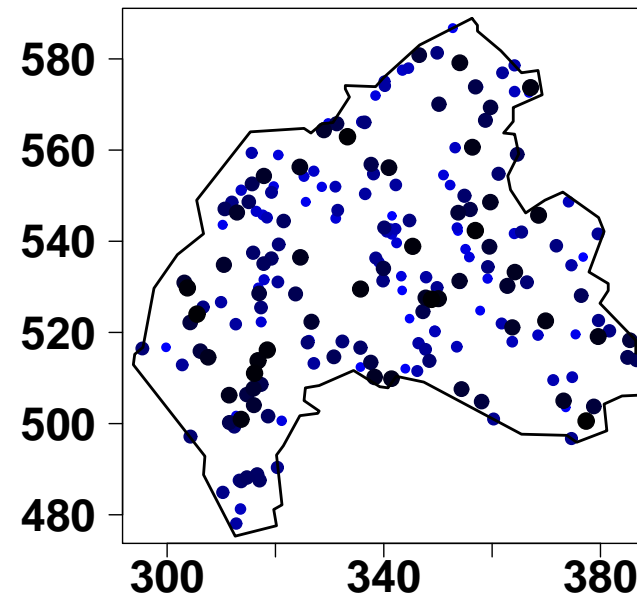
```
hpp1 <- rpp(lambda = 200, nsim = 5)
```

```
plot(hpp1$xyt[[2]], mark = TRUE)
```



```
hpp2 <- rpp(npoints=200, s.region=northcumbria,  
            t.region=c(1,500), discrete.time=TRUE)
```

```
plot(hpp2$xyt, mark = TRUE)
```



Processus de Poisson inhomogène

L'intensité λ du processus de Poisson homogène est remplacée par $\lambda(s, t)$.

★ Définition :

1. $N(\mathcal{D}_{ST})$ suit une loi de Poisson d'espérance $\int_S \int_T \lambda(s, t) dt ds$,
2. Sachant $N(\mathcal{D}_{ST}) = n$, les n points de \mathcal{D}_{ST} sont indépendants et de distribution $\lambda(s, t) / \int_S \int_T \lambda(s, t) dt ds$.

★ Simulation : via la technique d'effacement de points ("thinning").

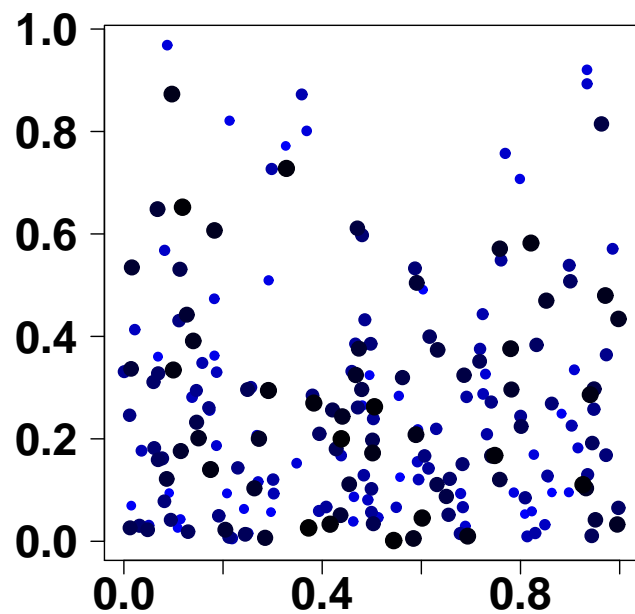
Supposons que pour tout x , $0 \leq \lambda(x) \leq c < \infty$.

1. Simuler un processus de Poisson homogène d'intensité c , Φ^* ,
2. Effacer indépendamment x_i de Φ^* avec la probabilité $p(x_i) = 1 - \frac{\lambda(x_i)}{c}$.

```

lbda1 <- function(x,y,t,a)
  {a*exp(-4*y) * exp(-2*t)}
ipp1 <- rpp(lambda=lbda1, npoints=200,
  a=1600/((1-exp(-4))*(1-exp(-2))))

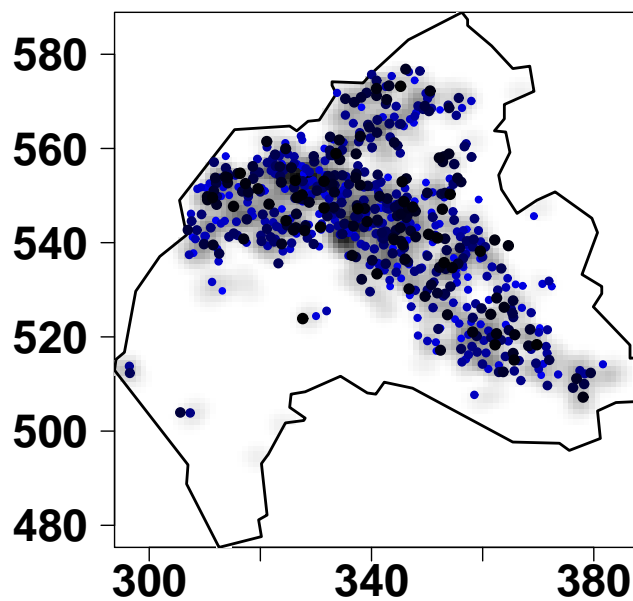
```



```

Ls <- kernel2d(as.points(fmd[,1:2]), h)
Lt <- dim(fmd)[1]*density(fmd[,3], n=200)$y
Lst <- Ls*z*Lt[k]/dim(fmd)[1]
ipp2 <- rpp(lambda="m", Lambda=Lst)

```



Processus de Cox

★ Définition :

1. $\{\Lambda(x) : x \in \mathbb{R}^2 \times \mathbb{R}\}$ est un processus aléatoire à valeurs non négatives,
2. Conditionnellement à $\{\Lambda(x) = \lambda(x) : x \in \mathbb{R}^2 \times \mathbb{R}\}$, les points forment un processus de Poisson d'intensité $\lambda(x)$.

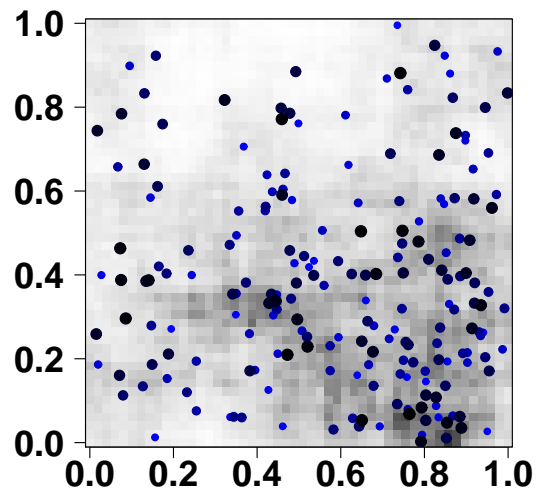
★ Processus de Cox log-Gaussien :

$\Lambda(x) = \exp(Z(x))$, où $Z(x)$ est un processus Gaussien.

★ Simulation :

1. Simuler un champ Gaussien $Z(x)$.
2. Définir $\lambda(x) = \exp(Z(x))$.
3. Procéder comme pour un processus de Poisson inhomogène.

```
lgcp <- rlgcp(npoints = 200, nx = 50, ny = 50, nt = 50, separable = FALSE,
             model = "gneiting", param = c(1,1,1,1,1,2), var.grf = 1, mean.grf = 0)
```



Modèles de covariance du processus Gaussien :

★ Covariance séparable : $c(h, t) = c_S(h)c_T(t)$, $h \in S, t \in T$

Modèles implémentés : exponentiel, Stable, Matérn, Cauchy,

★ Covariance non-séparable : $c(h, t) = \psi(t)^{-\alpha_6} \phi\left(\frac{h}{\psi(t)}\right)$,

où $\phi(r)$, $r \geq 0$ est une fonction monotone et $\psi(r)$, $r \geq 0$ est une fonction dont la dérivée est monotone.

Poisson Cluster Process

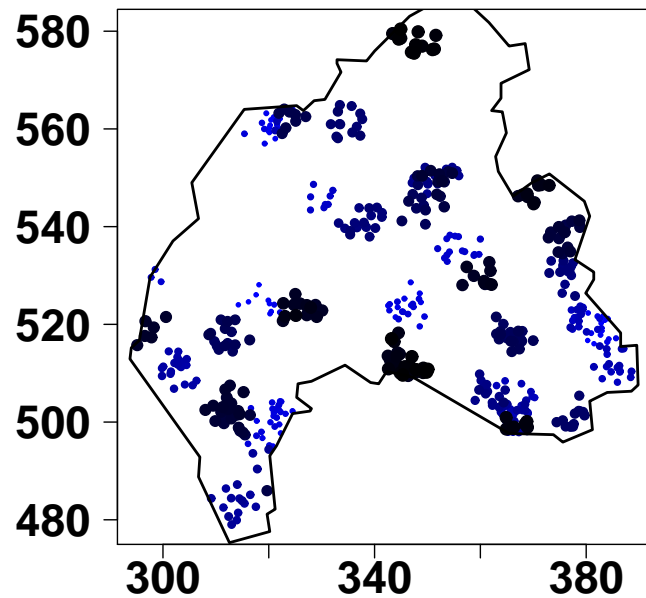
★ Définition :

1. Des points 'parents' sont issus d'un processus de Poisson d'intensité $\lambda_p(x)$.
2. Le nombre d'enfants par parent est une v.a. N_c d'espérance m_c , réalisée indépendamment pour chaque parent.
3. Les positions et temps des enfants sont distribués aléatoirement autour des parents.
4. Le processus final est constitué seulement des enfants.

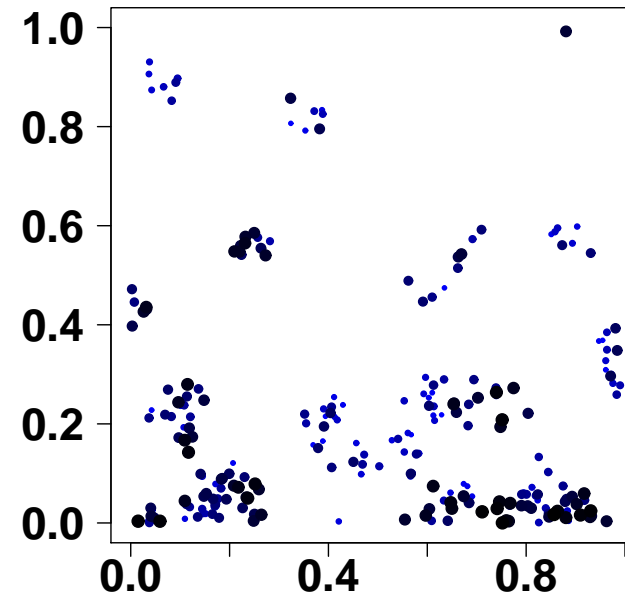
★ Exemples (cas spatial) :

- Processus de Neyman-Scott : $\lambda_p(x) = \lambda$,
- Processus de Thomas : les enfants sont distribués selon une loi Gaussienne.

```
pcp1 <- rpcp(nparents=50, mc=10,  
            cluster=c("normal","exponential"),  
            dispersion=c(5,5))  
plot(pcp1$xyt, mark=TRUE)
```



```
lbda <- function(x,y,t,a)  
        {a*exp(-4*y) * exp(-2*t)}  
pcp2 <- rpcp(nparents=50, npoints=250,  
            cluster="uniform", lambda=lbda)
```



Processus d'interaction

Processus d'inhibition : \rightsquigarrow propriétés de répulsion

★ Simple sequential inhibition process

1. s_1 et t_1 sont uniformément distribués dans S et T .

2. A l'étape k , $k = 2, \dots, m$:

s_k uniformément distribué sur $S \cap \{s : \|s - s_j\| \geq \delta_s, j = 1, \dots, k-1\}$

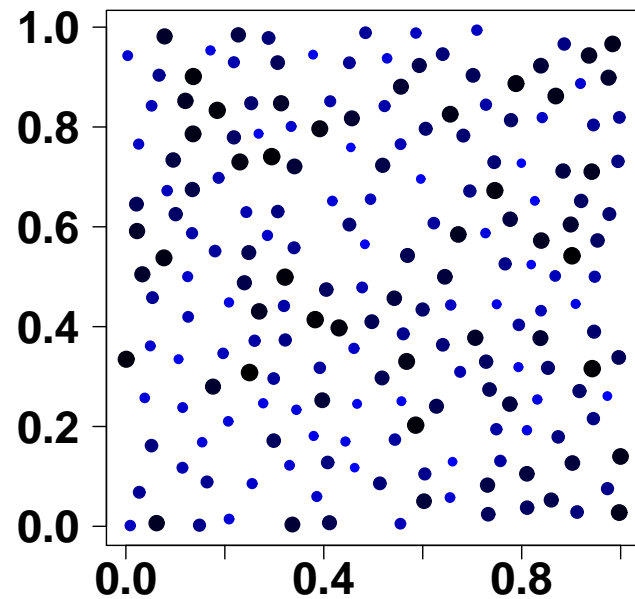
t_k uniformément distribué sur $T \cap \{t : |t - t_j| \geq \delta_t, j = 1, \dots, k-1\}$.

δ_s et δ_t : distances en-deçà desquelles il ne peut y avoir de points.

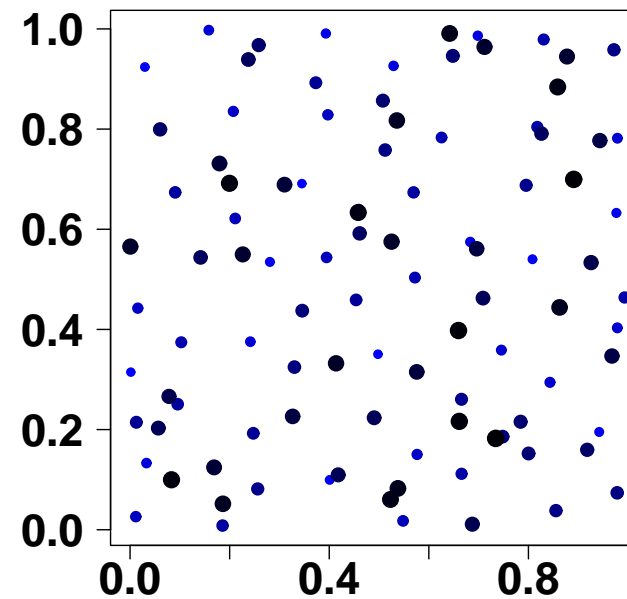
★ Modèle plus général :

On introduit dans 2. les probabilités $p(s, r)$ et $p(t, r)$ qu'un point potentiel (s, t) soit accepté dans le processus en fonction des r évènements précédents.

```
inh1 <- rinter(npoints=200, thetas=0,  
              deltas=0.05, thetat=0,  
              deltat=0.001, inhibition=TRUE)  
plot(inh1$xyt, mark=TRUE)
```



```
inh2 <- rinter(npoints=200, thetas=0.1,  
              deltas=0.05, thetat=0.1,  
              deltat=0.001, inhibition=TRUE)  
plot(inh2$xyt, mark=TRUE)
```



Processus de contagion : \rightsquigarrow propriétés d'attraction

★ Modèle simple :

1. s_1 et t_1 sont uniformément distribués dans S et T .

2. A l'étape k , $k = 2, \dots, m$:

s_k uniformément distribué sur $S \cap \{s : \|s - s_{k-1}\| \leq \delta_s\}$

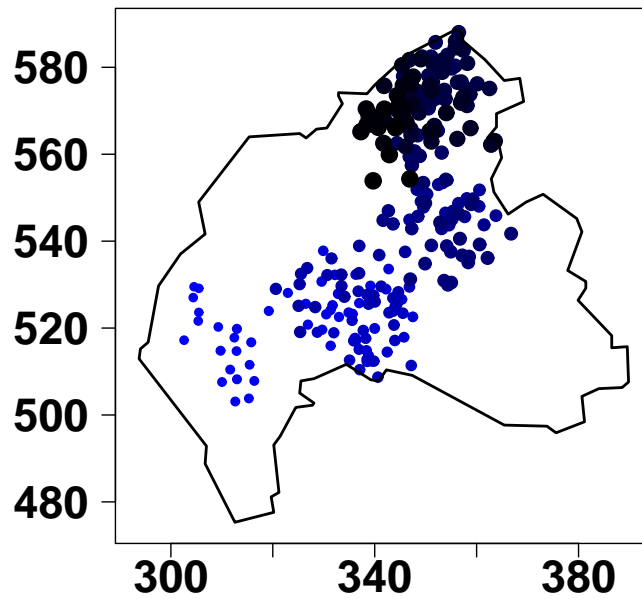
t_k uniformément distribué sur $T \cap \{t : |t - t_{k-1}| \leq \delta_t\}$.

δ_s et δ_t : distances au-delà desquelles il ne peut y avoir de points.

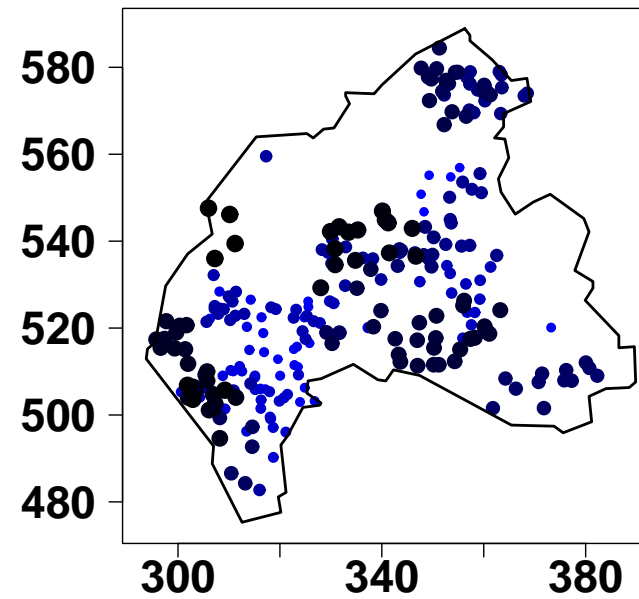
★ Modèle plus général :

On introduit dans 2. les probabilités $p(s, r)$ et $p(t, r)$ qu'un point potentiel (s, t) soit accepté dans le processus en fonction des r événements précédents.

```
cont1 <- rinter(npoints=250,  
  thetas=0, deltas=7.5,  
  thetat=0, deltat=10,  
  recent=1, inhibition=FALSE)  
plot(cont1$xyt, mark=TRUE)
```



```
cont2 <- rinter(npoints=250,  
  thetas=0.1, deltas=7.5,  
  thetat=0.1, deltat=10,  
  recent=1, inhibition=FALSE)  
plot(cont2$xyt, mark=TRUE)
```



Processus d'infection

A chaque individu à un temps t correspond un taux d'infection $h(t)$ qui dépend :

- d'une période de latence α ,
- d'un taux d'infection maximal β ,
- d'une période d'infection γ .

1. On choisit une localisation s_1 et un temps t_1 pour le premier évènement.

2. Sachant $\{(s_j, t_j), j = 1, \dots, k - 1\}$,

s_k est soit distribué symétriquement autour de s_{k-1}

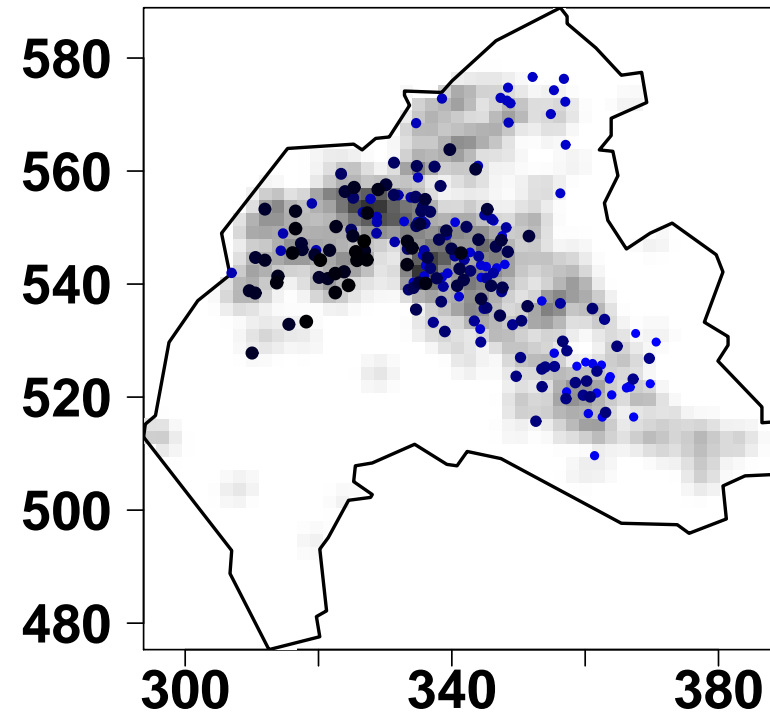
ou est distribué selon un processus de Poisson d'intensité $\lambda(s)$,

t_k est distribué exponentiellement ou uniformément autour de t_{k-1} .

Un point potentiel est accepté avec une probabilité

$$p_k = f(h(t|t_{k-1}, \alpha, \beta, \gamma)).$$

```
inf <- rinfec(npoints=200, alpha=4, beta=0.6, gamma=20, maxrad=c(12,20),  
s.distr="poisson", t.distr="uniform", recent=1, lambda=Ls$z,  
inhibition=FALSE)
```



Analyse de la structure des processus ponctuels

La fonction STIK (Space-time Inhomogeneous K function)

★ Basée sur les distances inter-événements.

★ Estimation :

$$\hat{K}(u, v) = \frac{1}{|S \times T|} \sum_{i=1}^n \sum_{j \neq i}^n \frac{1}{w_{ij}} \frac{1}{\lambda(x_i)\lambda(x_j)} \mathbf{1}_{\{\|s_i - s_j\| \leq u ; |t_i - t_j| \leq v\}},$$

avec w_{ij} : facteur correcteur des effets de bord,

u et v sont des distances (spatiales et temporelles respectivement).

★ Propriété : pour un processus de Poisson, $K(u, v) = 2\pi u^2 v$.

★ Implémenté dans `stpp` : fonction `STIKhat`.

\hat{K} pour mesurer l'agrégation spatio-temporelle

- $\hat{K}(u, v) - 2\pi u^2 v > 0 \Rightarrow$ agrégation,
Les points sont distribués en formant des groupes.
- $\hat{K}(u, v) - 2\pi u^2 v < 0 \Rightarrow$ régularité,
Les points sont distribués plus régulièrement que ne le produit le hasard.

\hat{K} pour mesurer l'interaction spatio-temporelle

En l'absence d'interaction, $\frac{K(u, v)}{K_S(u)K_T(v)}$ est constant.

\hat{K} pour valider des modèles de processus

- Simulation Monte Carlo,
- Enveloppes de confiance, tests de déviance.

Références

E. Gabriel, B. Rowlingson and P.J. Diggle (2012). STPP : An R package for plotting, simulating and analysing spatio-temporal point patterns. *Journal of Statistical Software*, submitted.

E. Gabriel and P.J. Diggle (2009). Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica*, 63 :43–51.