# Package 'stpp'

July 2, 2014

**Type** Package

**Title** Space-Time Point Pattern simulation, visualisation and analysis

**Version** 1.0-4

**Date** 2014-02-26

**Author** Edith Gabriel, Peter J Diggle, stan function by Barry Rowlingson

**Maintainer** Edith Gabriel <edith.gabriel@univ-avignon.fr>

**Depends** R (>= 2.10), splancs, KernSmooth, spatstat

**Suggests** rpanel, rgl

**Description** A package for analysing, simulating and displaying space-time point patterns

**License** GPL-3

**LazyLoad** yes

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2014-02-28 16:07:03

## R topics documented:

---

`animation` *Space-time data animation*

---

## Description

Provide an animation of spatio-temporal point patterns.

## Usage

```
animation(xyt, s.region, t.region, runtime=1, incident="red",
prevalent="pink3", pch=19, cex=0.5, plot.s.region=TRUE,
scales=TRUE, border.frac=0.05, add=FALSE)
```

## Arguments

| | |
|---|---|
| `xyt` | data-matrix containing the (x,y,t)-coordinates. |
| `s.region` | two-column matrix specifying polygonal region containing all data-locations `xyt[,1:2]`. If missing, `s.region` is the bounding box of `xyt[,1:2]`. |
| `t.region` | interval containing all data-times `xyt[,3]`. If missing, `t.region` is defined by the range of `xyt[,3]`. |
| `runtime` | approximate running time of animation, in seconds, but it is longer than expected. Can also be `NULL`. |
| `incident` | colour in which incident point `xyt[i,1:2]` is plotted at time `xyt[i,3]`. |
| `prevalent` | colour to which prevalent point `xyt[i,1:2]` fades at time `xyt[i+1,3]`. |
| `pch` | plotting symbol used for each point. |
| `cex` | magnification of plotting symbol relative to standard size. |
| `plot.s.region` | if `TRUE`, plot `s.region` as polygon. |
| `scales` | if `TRUE`, plot X and Y axes with scales. |
| `border.frac` | extent of border of plotting region surounding `s.region`, as fraction of ranges of X and Y. |
| `add` | if `TRUE`, add the animation to an existing plot. |

## Value

None

### Author(s)

Peter J Diggle, Edith Gabriel <edith.gabriel@univ-avignon.fr>.

---

`as.3dpoints` *Create data in spatio-temporal point format*

---

### Description

Create data in spatio-temporal point format.

### Usage

```
as.3dpoints(...)
```

### Arguments

| | |
|---|---|
| `...` | any object(s), such as x, y and t vectors of the same length, or a list or data frame containing x, y and t vectors. Valid options for ... are: a points object: returns it unaltered; a list with x, y and t elements of the same length: returns a points object with the x, y and t elements as the coordinates of the points; three vectors of equal length: returns a points object with the first vector as the x coordinates, the second vector as the y-coordinates and the third vector as the t-coordinates. |

### Value

The output is an object of the class `stpp`. as.3dpoints tries to return the argument(s) as a spatio-temporal points object.

### Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Peter Diggle, Barry Rowlingson.

---

`fmd` *2001 food-and-mouth epidemic, north Cumbria (UK)*

---

### Description

This data set gives the spatial locations and reported times of food-and-mouth disease in north Cumbria (UK), 2001. It is of no scientific value, as it deliberately excludes confidential information on farms at risk in the study-region. It is included in the package purely as an illustrative example.

### Usage

```
data(fmd)
```

## Format

A matrix containing (x,y,t) coordinates of the 648 observations.

## References

Diggle, P., Rowlingson, B. and Su, T. (2005). Point process methodology for on-line spatio-temporal disease surveillance. Environmetrics, 16, 423–34.

## See Also

`northcumbria` for boundaries of the county of north Cumbria.

---

`is.3dpoints` *Spatio-temporal point objects*

---

## Description

Tests for data in spatio-temporal point format.

## Usage

```
is.3dpoints(x)
```

## Arguments

| | |
|---|---|
| `x` | any object. |

## Value

is.3dpoints returns TRUE if x is a spatio-temporal points object, FALSE otherwise.

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Peter Diggle, Barry Rowlingson.

## northcumbria *Polygon boundary of north Cumbria*

### Description

This data set gives the boundary of the county of north Cumbria (UK).

### Usage

```
data(northcumbria)
```

### Format

A matrix containing (x,y) coordinates of the boundary.

### See Also

fmd for the space-time pattern of food-and-mouth disease in this county in 2001.

## PCFhat *Estimation of the space-time pair correlation function*

### Description

Compute an estimate of the space-time pair correlation function.

### Usage

```
PCFhat(xyt, s.region, t.region, dist, times, lambda,
ks="box", hs, kt="box", ht, correction = "isotropic")
```

### Arguments

| | |
|---|---|
| xyt | coordinates and times (x,y,t) of the point pattern. |
| s.region | two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the bounding box of xyt[,1:2] is considered. |
| t.region | vector containing the minimum and maximum values of the time interval. If t.region is missing, the range of xyt[,3] is considered. |
| dist | vector of distances u at which $g(u, v)$ is computed. |
| times | vector of times v at which $g(u, v)$ is computed. |
| lambda | vector of values of the space-time intensity function evaluated at the points (x,y,t) in SxT. If lambda is missing, the estimate of the space-time pair correlation function is computed considering $n/\|S \times T\|$ as an estimate of the space-time intensity. |

| | |
|---|---|
| ks | Kernel function for the spatial distances. Default is the "box" kernel. Can also be "epanech" for the Epanechnikov kernel or "gaussian" or "biweight". |
| hs | Bandwidth of the kernel function ks. |
| kt | Kernel function for the temporal distances. Default is the "box" kernel. Can also be "epanech" for the Epanechnikov kernel or "gaussian" or "biweight". |
| ht | Bandwidth of the kernel function kt. |
| correction | A character vector specifying the edge correction(s) to be applied among "isotropic", "border", "modified.border", "translate" and "none" (see Details). The default is "isotropic". |

## Details

An approximately unbiased estimator for the space-time pair correlation function, based on data giving the locations of events $x_i : i = 1, ...n$ on a spatio-temporal region $S \times T$, where S is an arbitrary polygon and T a time interval:

$$\widehat{g}(u,v) = \frac{1}{4\pi u} \sum_{i=1}^{n} \sum_{j \neq i} \frac{1}{w_{ij}} \frac{k_s(u - \|s_i - s_j\|) k_t(v - |t_i - t_j|)}{\lambda(x_i)\lambda(x_j)}$$

where $\lambda(x_i)$ is the intensity at $x_i = (s_i, t_i)$ and $w_{ij}$ is an edge correction factor to deal with spatial-temporal edge effects. The edge correction methods implemented are:

isotropic: $w_{ij} = |S \times T| w_{ij}^{(t)} w_{ij}^{(s)}$, where the temporal edge correction factor $w_{ij}^{(t)} = 1$ if both ends of the interval of length $2|t_i - t_j|$ centred at $t_i$ lie within T and $w_{ij}^{(t)} = 1/2$ otherwise and $w_{ij}^{(s)}$ is the proportion of the circumference of a circle centred at the location $s_i$ with radius $\|s_i - s_j\|$ lying in S (also called Ripley's edge correction factor).

border: $w_{ij} = \frac{\sum_{j=1}^{n} \mathbf{1}_{\{d(s_j,S)>u \ ; \ d(t_j,T)>v\}} / \lambda(x_j)}{\mathbf{1}_{\{d(s_i,S)>u \ ; \ d(t_i,T)>v\}}}$, where $d(s_i, S)$ denotes the distance between $s_i$ and the boundary of S and $d(t_i, T)$ the distance between $t_i$ and the boundary of T.

modified.border: $w_{ij} = \frac{|S_{\ominus u}| \times |T_{\ominus v}|}{\mathbf{1}_{\{d(s_i,S)>u \ ; \ d(t_i,T)>v\}}}$, where $S_{\ominus u}$ and $T_{\ominus v}$ are the eroded spatial and temporal region respectively, obtained by trimming off a margin of width u and v from the border of the original region.

translate: $w_{ij} = |S \cap S_{s_i - s_j}| \times |T \cap T_{t_i - t_j}|$, where $S_{s_i - s_j}$ and $T_{t_i - t_j}$ are the translated spatial and temporal regions.

none: No edge correction is performed and $w_{ij} = |S \times T|$.

$k_s()$ and $k_t()$ denotes kernel functions with bandwidth $h_s$ and $h_t$. Experience with pair correlation function estimation recommends box kernels (the default), see Illian et al. (2008). Epanechnikov, Gaussian and biweight kernels are also implemented. Whatever the kernel function, if the bandwidth is missing, a value is obtain from the function dpik of the package KernSmooth. Note that the bandwidths play an important role and their choice is crucial in the quality of the estimators as they heavily influence their variance.

## Value

A list containing:

| | |
|---|---|
| pcf | ndist x ntimes matrix containing values of $\hat{g}(u, v)$. |

| | |
|---|---|
| dist, times | parameters passed in argument. |
| kernel | a vector of names and bandwidths of the spatial and temporal kernels. |
| correction | the name(s) of the edge correction method(s) passed in argument. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>

## References

Gabriel E. (2014) Estimating second-order characteristics of inhomogeneous spatio-temporal point processes: influence of edge correction methods and intensity estimates. Methodology and computing in Applied Probabillity, 16(1).

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

Gabriel E., Rowlingson B., Diggle P. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software, 53(2), 1–29.

Baddeley A, Turner R (2000) Practical maximum pseudolikelihood for spatial point patterns. Aust NZ J Stat 42, 283–322.

Illian JB, Penttinen A, Stoyan H and Stoyan, D. (2008). Statistical Analysis and Modelling of Spatial Point Patterns. John Wiley and Sons, London.

## Examples

```
## Not run:
## First example
#################
data(fmd)
data(northcumbria)
FMD<-as.3dpoints(fmd[,1]/1000,fmd[,2]/1000,fmd[,3])
Northcumbria=northcumbria/1000

# estimation of the temporal intensity
Mt<-density(FMD[,3],n=1000)
mut<-Mt$y[findInterval(FMD[,3],Mt$x)]*dim(FMD)[1]

# estimation of the spatial intensity
h<-mse2d(as.points(FMD[,1:2]), Northcumbria, nsmse=50, range=4)
h<-h$h[which.min(h$mse)]
Ms<-kernel2d(as.points(FMD[,1:2]), Northcumbria, h, nx=5000, ny=5000)
atx<-findInterval(x=FMD[,1],vec=Ms$x)
aty<-findInterval(x=FMD[,2],vec=Ms$y)
mhat<-NULL
for(i in 1:length(atx)) mhat<-c(mhat,Ms$z[atx[i],aty[i]])

# estimation of the pair correlation function
g1 <- PCFhat(xyt=FMD, dist=1:20, times=1:20, lambda=mhat*mut/dim(FMD)[1],
 s.region=northcumbria/1000,t.region=c(1,200))
```

```
# plotting the estimation

plotPCF(g1)
plotPCF(g1,type="persp",theta=-65,phi=35)

## Second example
###################

xyt=rpp(lambda=200)
g2=PCFhat(xyt$xyt,dist=seq(0,0.16,by=0.02),
times=seq(0,0.16,by=0.02),correction=c("border","translate"))

plotPCF(g2,type="contour",which="border")


## End(Not run)
```

---

`plot.stpp` *Plot for spatio-temporal point objects*

---

## Description

This function plot either xy-locations and cumulative distribution of the times or xy-locations with time treated as a quantitative mark attached to each location.

## Usage

```
## S3 method for class 'stpp'
plot(x, s.region=NULL, t.region=NULL, mark=FALSE, mark.cexmin=0.4,
     mark.cexmax=1.2, mark.col=1, ...)
```

## Arguments

| | |
|---|---|
| `x` | any object of class `stpp` in spatio-temporal point format. |
| `s.region` | two-column matrix specifying polygonal region containing all data locations. If `s.region` is missing, the default limits are considered. |
| `t.region` | vector containing the minimum and maximum values of the time interval. If `t.region` is missing, the default limits are considered. |
| `mark` | Logical. If `FALSE` (default), xy-locations and cumulative distribution of the times are plotted. If `TRUE`, the time is treated as a quantitative mark attached to each location, and the locations are plotted with the size and/or colour of the plotting symbol determined by the value of the mark. |
| `mark.cexmin, mark.cexmax` | range of the size of the plotting symbol when `mark=TRUE`. |
| `mark.col` | colour of the plotting symbol when `mark=TRUE`. If `mark.col=0`, all locations have the same colour specified by the usual `col` argument. Otherwise, can be 1 or "black" (default), 2 or "red", 3 or "green", 4 or "blue", in which cases symbols colour is faded, and the darker corresponds to the most recent time. |
| `...` | further arguments to be passed to the function plot. Typical argument is `pch`. |

## Value

None

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>.

## References

Gabriel E., Rowlingson B., Diggle P. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software, 53(2), 1–29.

## See Also

`as.3dpoints` for creating data in spatio-temporal point format.

---

`plotK` *Plot the estimation of the Space-Time Inhomogeneous K-function*

---

## Description

Contour plot or perspective plot or image of the Space-Time Inhomogeneous K-function estimate.

## Usage

```
plotK(K,n=15,L=FALSE,type="contour",legend=TRUE,which=NULL,
main=NULL,...)
```

## Arguments

| | |
|---|---|
| `K` | result of the `STIKhat` function. |
| `n` | number of contour levels desired. |
| `L` | logical indicating whether $K_{ST}(u,v)$ or $L(u,v) = K_{ST}(u,v) - \pi u^2 v$ must be plotted. |
| `type` | specifies the kind of plot: `contour` by default, but can also be `persp` or `image` |
| `legend` | logical indicating whether a legend must be added to the plot. |
| `which` | A character specifying the edge correction among the ones used in `STIKhat`. If a single edge correction method was used in `STIKhat`, it is not necessary to specify `which`. |
| `main` | plot title. |
| `...` | additional arguments to `persp` if `persp=TRUE`, such as `theta` and `phi`. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>

## See Also

contour, persp, image and STIKhat for an example.

---

plotPCF — *Plot the estimation of the Space-Time Inhomogeneous pair correlation function*

---

## Description

Contour, image or perspective plot of the Space-Time Inhomogeneous pair correlation function estimate.

## Usage

```
plotPCF(PCF,n=15,type="contour",legend=TRUE,which=NULL,
main=NULL,...)
```

## Arguments

| | |
|---|---|
| PCF | result of the PCFhat function. |
| n | number of contour levels desired. |
| type | specifies the kind of plot: contour by default, but can also be persp or image |
| legend | logical indicating whether a legend must be added to the plot. |
| which | A character specifying the edge correction among the ones used in PCFhat. If a single edge correction method was used in PCFhat, it is not necessary to specify which. |
| main | plot title. |
| ... | additional arguments to persp if persp=TRUE, such as theta and phi. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>

## See Also

contour, persp, link{image} and PCFhat for an example.

rinfec *Generate infection point patterns*

## Description

Generate one (or several) realisation(s) of the infection process in a region S x T.

## Usage

```
rinfec(npoints, s.region, t.region, nsim=1, alpha, beta, gamma,
s.distr="exponential", t.distr="uniform", maxrad, delta, h="step",
g="min", recent=1, lambda=NULL, lmax=NULL, nx=100, ny=100, nt=1000,
t0, inhibition=FALSE, ...)
```

## Arguments

| | |
|---|---|
| npoints | number of points to simulate. |
| s.region | two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered. |
| t.region | vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval [0,1] is considered. |
| nsim | number of simulations to generate. Default is 1. |
| alpha | numerical value for the latent period. |
| beta | numerical value for the maximum infection rate. |
| gamma | numerical value for the infection period. |
| h | infection rate function which depends on alpha, beta and delta. Must be choosen among "step" and "gaussian". |
| s.distr | spatial distribution. Must be choosen among "uniform", "gaussian", "exponential" and "poisson". |
| t.distr | temporal distribution. Must be choosen among "uniform" and "exponential". |
| maxrad | single value or 2-vector of spatial and temporal maximum radiation respectively. If single value, the same value is used for space and time. |
| delta | spatial distance of inhibition/contagion. If missing, the spatial radiation is used |
| g | Compute the probability of acceptance of a new point from h and recent. Must be choosen among "min", "max" and "prod". |
| recent | If "all" consider all previous events. If is an integer, say N, consider only the N most recent events. |
| lambda | function or matrix defining the intensity of a Poisson process if s.distr is Poisson. |
| lmax | upper bound for the value of lambda. |
| nx, ny | define the 2-D grid on which the intensity is evaluated if s.distr is Poisson. |
| nt | used to discretize time to compute the infection rate function. |

| | |
|---|---|
| t0 | minimum time used to compute the infection rate function. Default is the minimum of t.region. |
| inhibition | Logical. If TRUE, an inhibition process is generated. Otherwise, it is a contagious process. |
| ... | additional parameters if lambda is a function. |

## Value

A list containing:

| | |
|---|---|
| xyt | matrix (or list of matrices if nsim>1) containing the points (x,y,t) of the simulated point pattern. xyt (or any element of the list if nsim>1) is an object of the class stpp. |
| s.region, t.region | parameters passed in argument. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Peter J Diggle.

## See Also

plot.stpp, animation and stan for plotting space-time point patterns.

## Examples

```
## Not run:
# inhibition; spatial distribution: uniform
inf1 = rinfec(npoints=100, alpha=0.2, beta=0.6, gamma=0.5,
maxrad=c(0.075,0.5), t.region=c(0,50), s.distr="uniform",
t.distr="uniform", h="step", p="min", recent="all",
inhibition=TRUE)
animation(inf1$xyt, cex=0.8, runtime=10)

# contagion; spatial distribution: Poisson with intensity a given matrix
data(fmd)
data(northcumbria)
h = mse2d(as.points(fmd[,1:2]), northcumbria, nsmse=30, range=3000)
h = h$h[which.min(h$mse)]
Ls = kernel2d(as.points(fmd[,1:2]), northcumbria, h, nx=50, ny=50)
inf2 = rinfec(npoints=100, alpha=4, beta=0.6, gamma=20, maxrad=c(12000,20),
s.region=northcumbria, t.region=c(1,2000), s.distr="poisson",
t.distr="uniform", h="step", p="min", recent=1,
lambda=Ls$z, inhibition=FALSE)


image(Ls$x, Ls$y, Ls$z, col=grey((1000:1)/1000)); polygon(northcumbria,lwd=2)
animation(inf2$xyt, add=TRUE, cex=0.7, runtime=15)

## End(Not run)
```

rinter *Generate interaction point patterns*

## Description

Generate one (or several) realisation(s) of the inhibition or contagious process in a region S x T.

## Usage

```
rinter(npoints,s.region,t.region,hs="step",gs="min",thetas=0,
deltas,ht="step",gt="min",thetat=1,deltat,recent="all",nsim=1,
discrete.time=FALSE,replace=FALSE,inhibition=TRUE,...)
```

## Arguments

| | |
|---|---|
| npoints | number of points to simulate. |
| s.region | two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered. |
| t.region | vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval [0,1] is considered. |
| hs, ht | function which depends on the distance between points and theta. Can be chosen among "step" and "gaussian" or can refer to a user defined function which only depend on d, theta, and delta (see details). If inhibition=TRUE, h is monotone, increasing, and must tend to 1 when the distance tends to infinity. $0 \leq$ h(d,theta) $\leq 1$. Otherwise, h is monotone, decreasing, and must tend to 1 when the distance tends to 0. |
| thetas, thetat | Parameters of hs and ht functions. |
| deltas, deltat | Spatial and temporal distance of inhibition. |
| gs, gt | Compute the probability of acceptance of a new point from hs or ht and recent. Must be choosen among "min", "max" and "prod". |
| recent | If "all" consider all previous events. If is an integer, say N, consider only the N most recent events. |
| nsim | number of simulations to generate. Default is 1. |
| discrete.time | if TRUE, times belong to $\mathbf{N}$, otherwise belong to $\mathbf{R}^+$. |
| replace | Logical. If TRUE allows times repeat. |
| inhibition | Logical. If TRUE, an inhibition process is generated. Otherwise, it is a contagious process. |
| ... | Additional parameters if hs and ht are defined by the user. |

## Value

A list containing:

`xyt` matrix (or list of matrices if `nsim`>1) containing the points (x,y,t) of the simulated point pattern. `xyt` (or any element of the list if `nsim`>1) is an object of the class `stpp`.

`s.region, t.region` parameters passed in argument.

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Peter J Diggle.

## See Also

`plot.stpp`, `animation` and `stan` for plotting space-time point patterns.

## Examples

```
# simple inhibition process
inh1 = rinter(npoints=200,thetas=0,deltas=0.05,thetat=0,deltat=0.001,
inhibition=TRUE)
## Not run: stan(inh1$xyt)

# inhibition process using hs and ht defined by the user
hs = function(d,theta,delta,mus=0.1)
{
 res=NULL
 a=(1-theta)/mus
 b=theta-a*delta
 for(i in 1:length(d))
{
if (d[i]<=delta) res=c(res,theta)
if (d[i]>(delta+mus)) res=c(res,1)
if (d[i]>delta & d[i]<=(delta+mus)) res=c(res,a*d[i]+b)
}
 return(res)
}
ht = function(d,theta,delta,mut=0.3)
{
 res=NULL
 a=(1-theta)/mut
 b=theta-a*delta
 for(i in 1:length(d))
{
if (d[i]<=delta) res=c(res,theta)
if (d[i]>(delta+mut)) res=c(res,1)
if (d[i]>delta & d[i]<=(delta+mut)) res=c(res,a*d[i]+b)
}
 return(res)
}
d=seq(0,1,length=100)
```

```
plot(d,hs(d,theta=0.2,delta=0.1,mus=0.1),xlab="",ylab="",type="l",
ylim=c(0,1),lwd=2,las=1)
lines(d,ht(d,theta=0.1,delta=0.05,mut=0.3),col=2,lwd=2)
legend("bottomright",col=1:2,lty=1,lwd=2,legend=c(expression(h[s]),
expression(h[t])),bty="n",cex=2)

inh2 = rinter(npoints=100, hs=hs, gs="min", thetas=0.2, deltas=0.1,
ht=ht, gt="min", thetat=0.1, deltat=0.05, inhibition=TRUE)
## Not run: animation(inh2$xyt, runtime=15, cex=0.8)

# simple contagious process for given spatial and temporal regions
data(northcumbria)
cont1 = rinter(npoints=250, s.region=northcumbria, t.region=c(1,200),
thetas=0, deltas=5000, thetat=0, deltat=10, recent=1, inhibition=FALSE)

## Not run:
plot(cont1$xyt,pch=19,s.region=cont1$s.region,mark=TRUE,mark.col=4)

## End(Not run)
```

---

`rlgcp` *Generate log-Gaussian Cox point patterns*

---

## Description

Generate one (or several) realisation(s) of the log-Gaussian cox process in a region S x T.

## Usage

```
rlgcp(s.region, t.region, replace=TRUE, npoints=NULL, nsim=1,
nx=100, ny=100, nt=100,separable=TRUE,model="exponential",
param=c(1,1,1,1,1,2), scale=c(1,1),var.grf=1,mean.grf=0,
lmax=NULL,discrete.time=FALSE,exact=FALSE)
```

## Arguments

| | |
|---|---|
| `s.region` | two-column matrix specifying polygonal region containing all data locations. If `s.region` is missing, the unit square is considered. |
| `t.region` | vector containing the minimum and maximum values of the time interval. If `t.region` is missing, the interval [0,1] is considered. |
| `npoints` | number of points to simulate. If NULL, the number of points is from a Poisson distribution with mean the double integral of lambda(s,t) over `s.region` and `t.region`. |
| `nsim` | number of simulations to generate. Default is 1. |
| `separable` | Logical. If TRUE, the covariance function of the Gaussian random field is separable. |

| | |
|---|---|
| `model` | vector of length 1 or 2 specifying the model(s) of covariance of the Gaussian random field. If `separable=TRUE` and `model` is of length 2, then the elements of `model` define the spatial and temporal covariances respectively. If `separable=TRUE` and `model` is of length 1, then the spatial and temporal covariances belongs to the same class of covariances, among "matern", "exponential", "stable", "cauchy" and "wave" (see Details). If `separable=FALSE`, `model` must be of length 1 and is either "gneiting" or "cesare" (see Details). |
| `param` | $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$. Vector of parameters of the covariance function (see Details). |
| `scale` | vector of length 2 defining the spatial and temporal scale. |
| `var.grf` | variance of the Gaussian random field. |
| `mean.grf` | mean of the Gaussian random field. |
| `replace` | logical allowing times repeat. |
| `nx,ny,nt` | define the size of the 3-D grid on which the intensity is evaluated. |
| `lmax` | upper bound for the value of $\lambda(x, y, t)$. |
| `discrete.time` | if TRUE, times belong to $\mathbf{N}$, otherwise belong to $\mathbf{R}^+$. |
| `exact` | logical allowing exact simulation of Gaussian random fields (see manual for details). |

## Details

We implemented stationary, isotropic spatio-temporal covariance functions.

*Separable covariance functions*

$$c(h,t) = c_s(\|h\|)\, c_t(|t|), h \in S, t \in T$$

The purely spatial and purely temporal covariance functions can be:

- Exponential: $c(r) = \exp(-r)$,
- Stable: $c(r) = \exp(-r^\alpha)$, $\alpha \in [0, 2]$,
- Cauchy: $c(r) = (1 + r^2)^{-\alpha}$, $\alpha > 0$,
- Wave: $c(r) = \frac{\sin(r)}{r}$ if $r > 0$, $c(0) = 1$,
- Matern: $c(r) = \frac{(\alpha r)^\nu}{2^{\nu-1}\Gamma(\nu)}\mathcal{K}_\nu(\alpha r)$, $\nu > 0$ and $\alpha > 0$.
  $\mathcal{K}_\nu$ is the modified Bessel function of second kind:

$$\mathcal{K}_\nu(x) = \frac{\pi}{2}\frac{I_{-\nu}(x) - I_\nu(x)}{\sin(\pi\nu)},$$

  with $I_\nu(x) = \left(\frac{x}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{1}{k!\Gamma(\nu+k+1)} \left(\frac{x}{2}\right)^{2k}$.
  The parameters $\alpha_1$ and $\alpha_2$ correspond to the parameters of the spatial and temporal covariance respectively. For the Matern model, the parameters $\alpha_1$, $\alpha_3$ and $\alpha_2$, $\alpha_4$ correspond to the parameters $\nu$, $\alpha$ of the spatial and temporal covariance.

*Non-separable covariance functions*

The spatio-temporal covariance function can be:

- gneiting: $c(h,t) = \psi(t/\beta_2)^{-\alpha_6}\phi\left(\frac{h/\beta_1}{\psi(t/\beta_2)}\right)$, $\beta_1, \beta_2 > 0$,
  - If $\alpha_2 = 1$, $\phi(r)$ is the Stable model.
  - if $\alpha_2 = 2$, $\phi(r)$ is the Cauchy model.
  - If $\alpha_2 = 3$, $\phi(r)$ is the Matern model.
  - If $\alpha_5 = 1$, $\psi^2(r) = (r^{\alpha_3} + 1)^{\alpha_4}$,
  - If $\alpha_5 = 2$, $\psi^2(r) = (\alpha_4^{-1} r^{\alpha_3} + 1)/(r^{\alpha_3} + 1)$,
  - If $\alpha_5 = 3$, $\psi^2(r) = -\log(r^{\alpha_3} + 1/\alpha_4)/\log\alpha_4$,

  The parameter $\alpha_1$ is the respective parameter for the model of $\phi(\cdot)$, $\alpha_3 \in (0, 2]$, $\alpha_4 \in (0, 1]$ and $\alpha_6 \geq 2$.
- cesare: $c(h,t) = (1 + (h/\beta_1)^{\alpha_1} + (t/\beta_2)^{\alpha_2})^{-\alpha_3}$, $\beta_1, \beta_2 > 0$, $\alpha_1, \alpha_2 \in [1, 2]$ and $\alpha_3 \geq 3/2$.

## Value

A list containing:

| | |
|---|---|
| `xyt` | matrix (or list of matrices if `nsim`>1) containing the points (x,y,t) of the simulated point pattern. `xyt` (or any element of the list if `nsim`>1) is an object of the class `stpp`. |
| `s.region, t.region` | parameters passed in argument. |
| `Lambda` | nx * ny * nt array (or list of array if `nsim`>1) of the intensity. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Peter J Diggle.

## References

Chan, G. and Wood A. (1997). An algorithm for simulating stationary Gaussian random fields. Applied Statistics, Algorithm Section, 46, 171–181.

Chan, G. and Wood A. (1999). Simulation of stationary Gaussian vector fields. Statistics and Computing, 9, 265–268.

Gneiting T. (2002). Nonseparable, stationary covariance functions for space-time data. Journal of the American Statistical Association, 97, 590–600.

## See Also

`plot.stpp`, `animation` and `stan` for plotting space-time point patterns.

## Examples

```
# non separable covariance function:
lgcp1 <- 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)
N <- lgcp1$Lambda[,,1];for(j in 2:(dim(lgcp1$Lambda)[3])){N <-
N+lgcp1$Lambda[,,j]}
## Not run:
```

```
image(N,col=grey((1000:1)/1000));box()
animation(lgcp1$xyt, cex=0.8, runtime=10, add=TRUE, prevalent="orange")

## End(Not run)
# separable covariance function:
lgcp2 <- rlgcp(npoints=200, nx=50, ny=50, nt=50, separable=TRUE,
model="exponential", param=c(1,1,1,1,1,2), var.grf=2, mean.grf=-0.5*2)
N <- lgcp2$Lambda[,,1];for(j in 2:(dim(lgcp2$Lambda)[3])){N <-
N+lgcp2$Lambda[,,j]}
## Not run:
image(N,col=grey((1000:1)/1000));box()
animation(lgcp2$xyt, cex=0.8, pch=20, runtime=10, add=TRUE,
prevalent="orange")

## End(Not run)
```

---

rpcp *Generate Poisson cluster point patterns*

---

## Description

Generate one (or several) realisation(s) of the Poisson cluster process in a region S x T.

## Usage

```
rpcp(s.region, t.region, nparents=NULL, npoints=NULL, lambda=NULL,
mc=NULL, nsim=1, cluster="uniform", dispersion, infectious=TRUE,
edge = "larger.region", larger.region=larger.region, tronc=1,...)
```

## Arguments

| | |
|---|---|
| s.region | two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered. |
| t.region | vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval [0,1] is considered. |
| nparents | number of parents. If NULL, nparents is from a Poisson distribution with intensity lambda. |
| npoints | number of points to simulate. If NULL (default), the number of points is from a Poisson distribution with mean the double integral of the intensity over s.region and t.region. |
| lambda | intensity of the parent process. Can be either a numeric value, a function, or a 3d-array (see rpp). If NULL, it is constant and equal to nparents / volume of the domain. |
| mc | average number of children per parent. It is used when npoints is NULL. |
| nsim | number of simulations to generate. |

| | |
|---|---|
| `cluster` | distribution of children: "uniform", "normal" and "exponential" are currently implemented. Either a single value if the distribution in space and time is the same, or a vector of length 2, giving first the spatial distribution of children and then the temporal distribution. |
| `dispersion` | scale parameter. It equals twice the standard deviation of location of children relative to their parent for a normal distribution of children; the mean for an exponential distribution and half range for an uniform distribution. |
| `infectious` | If TRUE, offspring's times are always greater than parent's time). |
| `edge` | specify the edge correction to use "larger.region" or "without". |
| `larger.region` | By default, the larger spatial region is the convex hull of `s.region` enlarged by the spatial related value of `dispersion` and the larger time interval is `t.region` enlarged by the temporal related value of `dispersion`. One can over-ride default using the 2-vector parameter `larger.region`. |
| `tronc` | parameter of the truncated exponential distribution for the distribution of children. |
| `...` | additional parameters of the intensity of the parent process. |

## Value

A list containing:

| | |
|---|---|
| `xyt` | matrix (or list of matrices if `nsim`>1) containing the points (x,y,t) of the simulated point pattern. `xyt` (or any element of the list if `nsim`>1) is an object of the class `stpp`. |
| `s.region, t.region` | parameters passed in argument. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Peter J Diggle.

## See Also

`plot.stpp`, `animation` and `stan` for plotting space-time point patterns.

## Examples

```
# homogeneous Poisson distribution of parents

data(northcumbria)
pcp1 <- rpcp(nparents=50, npoints=500, s.region=northcumbria,
t.region=c(1,365), cluster=c("normal","exponential"),
maxrad=c(5000,5))
## Not run:
animation(pcp1$xyt, s.region=pcp1$s.region, t.region=pcp1$t.region,
runtime=5)

## End(Not run)
# inhomogeneous Poisson distribution of parents
```

```
lbda <- function(x,y,t,a){a*exp(-4*y) * exp(-2*t)}
pcp2 <- rpcp(nparents=50, npoints=500, cluster="normal", lambda=lbda,
a=4000/((1-exp(-4))*(1-exp(-2))))
## Not run:
stan(pcp2$xyt)

## End(Not run)
```

---

rpp *Generate Poisson point patterns*

---

## Description

Generate one (or several) realisation(s) of the (homogeneous or inhomogeneous) Poisson process in a region S x T.

## Usage

```
rpp(lambda, s.region, t.region, npoints=NULL, nsim=1, replace=TRUE,
    discrete.time=FALSE, nx=100, ny=100, nt=100, lmax=NULL, ...)
```

## Arguments

| | |
|---|---|
| lambda | Spatio-temporal intensity of the Poisson process. If lambda is a single positive number, the function generates realisations of a homogeneous Poisson process, whilst if lambda is a function of the form $\lambda(x, y, t, \ldots)$ or a 3D-array it generates realisations of an inhomogeneous Poisson process. |
| s.region | two-column matrix specifying polygonal region containing all data locations. If s.region is missing, the unit square is considered. |
| t.region | vector containing the minimum and maximum values of the time interval. If t.region is missing, the interval [0,1] is considered. |
| replace | logical allowing times repeat. |
| npoints | number of points to simulate. If NULL, the number of points is from a Poisson distribution with mean the double integral of lambda over s.region and t.region. |
| discrete.time | if TRUE, times belong to $\mathbf{N}$, otherwise belong to $\mathbf{R}^+$. |
| nsim | number of simulations to generate. Default is 1. |
| nx,ny,nt | define the size of the 3-D grid on which the intensity is evaluated. |
| lmax | upper bound for the value of $\lambda(x, y, t)$, if lambda is a function. |
| ... | additional parameters if lambda is a function. |

## Value

A list containing:

| | |
|---|---|
| `xyt` | matrix (or list of matrices if `nsim`>1) containing the points (x,y,t) of the simulated point pattern. `xyt` (or any element of the list if `nsim`>1) is an object of the class `stpp`. |
| `t.index` | vector of times index. |
| `Lambda` | nx x ny x nt array of the intensity surface at each time. |
| `s.region, t.region, lambda` | parameters passed in argument. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr> and Peter J Diggle.

## See Also

`plot.stpp`, `animation` and `stan` for plotting space-time point patterns.

## Examples

```
# Homogeneous Poisson process
# ---------------------------
hpp1 <- rpp(lambda=200,replace=FALSE)

## Not run: stan(hpp1$xyt)

# fixed number of points, discrete time, with time repeat.
data(northcumbria)
hpp2 = rpp(npoints=500, s.region=northcumbria, t.region=c(1,1000),
discrete.time=TRUE)
## Not run:
polymap(northcumbria)
animation(hpp2$xyt, s.region=hpp2$s.region, t.region=hpp2$t.region,
runtime=10, add=TRUE)

## End(Not run)
# Inhomogeneous Poisson process
# -----------------------------

# intensity defined by a function
lbda1 = function(x,y,t,a){a*exp(-4*y) * exp(-2*t)}
ipp1 = rpp(lambda=lbda1, npoints=400, a=3200/((1-exp(-4))*(1-exp(-2))))
## Not run: stan(ipp1$xyt)

# intensity defined by a matrix
data(fmd)
data(northcumbria)
h = mse2d(as.points(fmd[,1:2]), northcumbria, nsmse=30, range=3000)
h = h$h[which.min(h$mse)]
Ls = kernel2d(as.points(fmd[,1:2]), northcumbria, h, nx=100, ny=100)
```

```
Lt = dim(fmd)[1]*density(fmd[,3], n=200)$y
Lst=array(0,dim=c(100,100,200))
for(k in 1:200) Lst[,,k] <- Ls$z*Lt[k]/dim(fmd)[1]
ipp2 = rpp(lambda=Lst, s.region=northcumbria, t.region=c(1,200),
discrete.time=TRUE)
## Not run:
par(mfrow=c(1,1))
image(Ls$x, Ls$y, Ls$z, col=grey((1000:1)/1000)); polygon(northcumbria)
animation(ipp2$xyt, add=TRUE, cex=0.5, runtime=15)

## End(Not run)
```

---

`sim.stpp` *Generate spatio-temporal point patterns*

---

## Description

Generate one (or several) realisation(s) of a spatio-temporal point process in a region S x T.

## Usage

```
sim.stpp(class="poisson", s.region, t.region, npoints=NULL,
nsim=1, ...)
```

## Arguments

| | |
|---|---|
| `class` | Must be chosen among "poisson", "cluster", "cox", "infectious" and "inhibition". |
| `s.region` | two-column matrix specifying polygonal region containing all data locations. If `s.region` is missing, the unit square is considered. |
| `t.region` | vector containing the minimum and maximum values of the time interval. If `t.region` is missing, the interval [0,1] is considered. |
| `npoints` | number of points to simulate. |
| `nsim` | number of simulations to generate. Default is 1. |
| `...` | additional parameters related to the `class` parameter. See `rpp` for the Poisson process; `rpcp` for the Poisson cluster process; `rlgcp` for the Log-Gaussian Cox process; `rinter` for the interaction (inhibition or contagious) process and `rinfec` for the infectious process. |

## Value

A list containing:

| | |
|---|---|
| `xyt` | matrix (or list of matrices if `nsim`>1) containing the points (x,y,t) of the simulated point pattern. `xyt` (or any element of the list if `nsim`>1) is an object of the class `stpp`. |
| `s.region, t.region` | parameters passed in argument. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>

## See Also

rpp, rinfec, rinter, rpcp and rlgcp for the simulation of Poisson, infectious, interaction, Poisson cluster and log-gaussian Cox processes respectively; and plot.stpp, animation and stan for plotting space-time point patterns.

---

stan *(3D) space-time data animation*

---

## Description

Displays (x,y,t) point data and enables dynamic highlighting of time slices.

## Usage

```
stan(xyt,tlim=range(xyt[,3],na.rm=TRUE),twid=diff(tlim)/20,
persist=FALSE,states,bgpoly,bgframe=TRUE,bgimage,
bgcol=gray(seq(0,1,len=12)),axes=TRUE)
```

## Arguments

| | |
|---|---|
| xyt | A 3-column matrix of x,y,t coordinates |
| tlim | A two-element vector of upper and lower time limits |
| twid | The initial time window width |
| persist | Whether to display points before time window |
| states | How to display points - see Details |
| bgpoly | A polygon to draw on the background plane |
| bgframe | Whether to extend the bgpoly to the front plane |
| bgimage | An list with x,y vectors and z matrix to display on the background plane |
| bgcol | A colour palette vector with which to draw the bgimage |
| axes | logical value indicating whether labels should be added. |

## Details

This function requires the rpanel and rgl packages. It uses rpanel for the sliders to control the graphics, and rgl for its ability to do flicker-free graphics.

The sliders set the position and width of the temporal highlight window. For 'time' slider set to time T and 'width' slider set to W, highlighted points are those with time coordinate t such that T-W < t < T.

How points are shown is configured with the states parameter. This is a list of length 3 specifying how points before the time window, inside the time window, and after the time window are

displayed. Each element is a list of parameters as would be passed to material3d() together with a radius element. Points are drawn as spheres with the corresponding material and radius as a fraction of the spatial span of the data.

By default the third state is invisible, and the first two states are different. By calling with the default for states and persist=TRUE, then the first state is set to the same as the second state. This has the effect of showing all points at time < T with the same sphere type.

If the user specifies the states parameter, then persist is ignored. The user can emulate the persist behaviour by specifying a states list with identical parameters for states 1 and 2.

Note that each state element should specify all material3d parameters used in any of the state elements. This is to make sure the parameters are reset for each of the sets of points.

The background polygon must be a simple 2-column vector of x and y coordinates. When used with bgframe=TRUE, the polygon is also drawn on the front plane, and the convex hull points are connected front to back in order to visualise the space-time prism that the data are contained in.

A raster image can be displayed on the back plane by setting the bgimage parameter. This must be a list with x, y and z components as needed by the image function. Note that x and y define the center of cells and so must be the same length as the dimensions of z - the image function can accept x and y values that are one longer than the dimensions of z to define the edges, but bgimage does not allow that.

## Value

A list of the slider parameters when the dialog is quitted.

## Author(s)

Barry Rowlingson <b.rowlingson@lancaster.ac.uk>, Edith Gabriel

---

# STIKhat *Estimation of the Space-Time Inhomogeneous K-function*

---

## Description

Compute an estimate of the Space-Time Inhomogeneous K-function.

## Usage

```
STIKhat(xyt, s.region, t.region, dist, times, lambda,
correction="isotropic", infectious=FALSE)
```

## Arguments

| | |
|---|---|
| `xyt` | coordinates and times (x,y,t) of the point pattern. |
| `s.region` | two-column matrix specifying polygonal region containing all data locations. If `s.region` is missing, the bounding box of `xyt[,1:2]` is considered. |
| `t.region` | vector containing the minimum and maximum values of the time interval. If `t.region` is missing, the range of `xyt[,3]` is considered. |

| | |
|---|---|
| `dist` | vector of distances u at which $K(u,v)$ is computed. If missing, the maximum of `dist` is given by min(Sx,Sy)/4, where Sx and Sy represent the maximum width and height of `s.region`. |
| `times` | vector of times v at which $K(u,v)$ is computed. |
| `lambda` | vector of values of the space-time intensity function evaluated at the points (x,y,t) in SxT. If `lambda` is missing, the estimate of the space-time K-function is computed as for the homogeneous case (Diggle et al., 1995), i.e. considering $n/\|S \times T\|$ as an estimate of the space-time intensity. |
| `correction` | A character vector specifying the edge correction(s) to be applied among `"isotropic"`, `"border"`, `"modified.border"`, `"translate"` and `"none"` (see Details). The default is `"isotropic"`. |
| `infectious` | logical value. If `TRUE`, only future events are considered and the isotropic edge correction method is used. See Details. |

## Details

Gabriel (2014) proposes the following unbiased estimator for the STIK-function, based on data giving the locations of events $x_i : i = 1, ...n$ on a spatio-temporal region $S \times T$, where S is an arbitrary polygon and T is a time interval:

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

where $\lambda(x_i)$ is the intensity at $x_i = (s_i, t_i)$ and $w_{ij}$ is an edge correction factor to deal with spatial-temporal edge effects. The edge correction methods implemented are:

`isotropic`: $w_{ij} = |S \times T| w_{ij}^{(t)} w_{ij}^{(s)}$, where the temporal edge correction factor $w_{ij}^{(t)} = 1$ if both ends of the interval of length $2|t_i - t_j|$ centred at $t_i$ lie within T and $w_{ij}^{(t)} = 1/2$ otherwise and $w_{ij}^{(s)}$ is the proportion of the circumference of a circle centred at the location $s_i$ with radius $\|s_i - s_j\|$ lying in S (also called Ripley's edge correction factor).

`border`: $w_{ij} = \frac{\sum_{j=1}^{n} \mathbf{1}\{d(s_j,S)>u \ ; \ d(t_j,T)>v\}/\lambda(x_j)}{\mathbf{1}_{\{d(s_i,S)>u \ ; \ d(t_i,T)>v\}}}$, where $d(s_i, S)$ denotes the distance between $s_i$ and the boundary of S and $d(t_i, T)$ the distance between $t_i$ and the boundary of T.

`modified.border`: $w_{ij} = \frac{|S_{\ominus u}| \times |T_{\ominus v}|}{\mathbf{1}_{\{d(s_i,S)>u \ ; \ d(t_i,T)>v\}}}$, where $S_{\ominus u}$ and $T_{\ominus v}$ are the eroded spatial and temporal region respectively, obtained by trimming off a margin of width u and v from the border of the original region.

`translate`: $w_{ij} = |S \cap S_{s_i - s_j}| \times |T \cap T_{t_i - t_j}|$, where $S_{s_i - s_j}$ and $T_{t_i - t_j}$ are the translated spatial and temporal regions.

`none`: No edge correction is performed and $w_{ij} = |S \times T|$.

If parameter `infectious = TRUE`, ony future events are considered and the estimator is, using an isotropic edge correction factor (Gabriel and Diggle, 2009):

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

In this equation, the points $x_i = (s_i, t_i)$ are ordered so that $t_i < t_{i+1}$, with ties due to round-off error broken by randomly unrounding if necessary. To deal with temporal edge-effects, for each v,

$n_v$ denotes the number of events for which $t_i \leq T_1 - v$, with $T = [T_0, T_1]$. To deal with spatial edge-effects, we use Ripley's method.

If `lambda` is missing in argument, `STIKhat` computes an estimate of the space-time (homogeneous) K-function:

$$\widehat{K}(u,v) = \frac{|S \times T|}{n_v(n-1)} \sum_{i=1}^{n_v} \sum_{j=1;j>i}^{n_v} \frac{1}{w_{ij}} \mathbf{1}_{\{u_{ij} \leq u\}} \mathbf{1}_{\{t_j - t_i \leq v\}}$$

## Value

A list containing:

| | |
|---|---|
| `Khat` | ndist x ntimes matrix containing values of $\hat{K}_{ST}(u,v)$. |
| `Ktheo` | ndist x ntimes matrix containing theoretical values for a Poisson process; $\pi u^2 v$ for K and $2\pi u^2 v$) for K^*. |
| `dist, times, infectious` | parameters passed in argument. |
| `correction` | the name(s) of the edge correction method(s) passed in argument. |

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>

## References

Gabriel E. (2014) Estimating second-order characteristics of inhomogeneous spatio-temporal point processes: influence of edge correction methods and intensity estimates. Methodology and computing in Applied Probabillity, 16(1).

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

Gabriel E., Rowlingson B., Diggle P. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software, 53(2), 1–29.

Baddeley A, Turner R (2000) Practical maximum pseudolikelihood for spatial point patterns. Aust NZ J Stat 42, 283–322.

Baddeley A., Moller J. and Waagepetersen R. (2000). Non- and semi-parametric estimation of interaction in inhomogeneous point patterns. Statistica Neerlandica, 54, 329–350.

Diggle P. , Chedwynd A., Haggkvist R. and Morris S. (1995). Second-order analysis of space-time clustering. Statistical Methods in Medical Research, 4, 124–136.

## Examples

```
## Not run:
## First example
#################

data(fmd)
data(northcumbria)
FMD<-as.3dpoints(fmd[,1]/1000,fmd[,2]/1000,fmd[,3])
```

```
Northcumbria=northcumbria/1000

# estimation of the temporal intensity
Mt<-density(FMD[,3],n=1000)
mut<-Mt$y[findInterval(FMD[,3],Mt$x)]*dim(FMD)[1]

# estimation of the spatial intensity
h<-mse2d(as.points(FMD[,1:2]), Northcumbria, nsmse=50, range=4)
h<-h$h[which.min(h$mse)]
Ms<-kernel2d(as.points(FMD[,1:2]), Northcumbria, h, nx=5000, ny=5000)
atx<-findInterval(x=FMD[,1],vec=Ms$x)
aty<-findInterval(x=FMD[,2],vec=Ms$y)
mhat<-NULL
for(i in 1:length(atx)) mhat<-c(mhat,Ms$z[atx[i],aty[i]])

# estimation of the STIK function
u <- seq(0,10,by=1)
v <- seq(0,15,by=1)
stik1 <- STIKhat(xyt=FMD, s.region=northcumbria/1000,t.region=c(1,200),
lambda=mhat*mut/dim(FMD)[1], dist=u, times=v, infectious=TRUE)

# plotting the estimation
plotK(stik1)
plotK(stik1,type="persp",theta=-65,phi=35)

## Second example
##################

xyt=rpp(lambda=200)
stik2=STIKhat(xyt$xyt,dist=seq(0,0.16,by=0.02),
times=seq(0,0.16,by=0.02),correction=c("border","translate"))

plotK(stik2,type="contour",legend=T,which="translate")

## End(Not run)
```

---

stpp *Space-Time Point Pattern simulation, visualisation and analysis*

---

## Description

This package provides models of spatio-temporal point processes in a region S x T and statistical tools for analysing second-order properties of such processes. It also includes static and dynamic (2D and 3D) plots. stpp is the first dedicated unified computational environment in the area of spatio-temporal point processes.

The stpp package depends upon some other packages:

splancs: spatial and space-time point pattern analysis

rgl: interactive 3D plotting of densities and surfaces

`rpanel`: simple interactive controls for R using `tcltk` package

`KernSmooth`: functions for kernel smoothing for Wand & Jones (1995)

## Details

`stpp` is a package for simulating, analysing and visualising patterns of points in space and time.

Following is a summary of the main functions and the dataset in the `stpp` package.

*To visualise a spatio-temporal point pattern*

- `animation`: space-time data animation.
- `as.3dpoints`: create data in spatio-temporal point format.
- `plot.stpp`: plot spatio-temporal point object. Either a two-panels plot showing spatial locations and cumulative times, or a one-panel plot showing spatial locations with times treated as a quantitative mark attached to each location.
- `stan`: 3D space-time animation.

*To simulate spatio-temporal point patterns*

- `rinfec`: simulate an infection point process,
- `rinter`: simulate an interaction (inhibition or contagious) point process,
- `rlgcp`: simulate a log-Gaussian Cox point process,
- `rpcp`: simulate a Poisson cluster point process,
- `rpp`: simulate a Poisson point process.

*To analyse spatio-temporal point patterns*

- `PCFhat`: space-time inhomogeneous pair correlation function,
- `STIKhat`: space-time inhomogeneous K-function.

*Dataset*

`fmd`: 2001 food-and-mouth epidemic in north Cumbria (UK).

## Author(s)

Edith Gabriel <edith.gabriel@univ-avignon.fr>, Barry Rowlingson and Peter J Diggle.

## References

Baddeley A., Moller J. and Waagepetersen R. (2000). Non- and semi-parametric estimation of interaction in inhomogeneous point patterns. Statistica Neerlandica, 54, 329–350.

Chan, G. and Wood A. (1997). An algorithm for simulating stationary Gaussian random fields. Applied Statistics, Algorithm Section, 46, 171–181.

Chan, G. and Wood A. (1999). Simulation of stationary Gaussian vector fields. Statistics and Computing, 9, 265–268.

Diggle P. , Chedwynd A., Haggkvist R. and Morris S. (1995). Second-order analysis of space-time clustering. Statistical Methods in Medical Research, 4, 124–136.

Gabriel E. (2014) Estimating second-order characteristics of inhomogeneous spatio-temporal point processes: influence of edge correction methods and intensity estimates. Methodology and computing in Applied Probabillity, 16(1).

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

Gabriel E., Rowlingson B., Diggle P. (2013) stpp: an R package for plotting, simulating and analyzing Spatio-Temporal Point Patterns. Journal of Statistical Software, 53(2), 1–29.

Gneiting T. (2002). Nonseparable, stationary covariance functions for space-time data. Journal of the American Statistical Association, 97, 590–600.

# Index