Package 'stlnpp'

May 27, 2024

Type Package

Version 0.4.0

Title Spatio-Temporal Analysis of Point Patterns on Linear Networks

Maintainer Mehdi Moradi <m2.moradi@yahoo.com>

Description Statistical analysis of spatio-temporal point processes on linear networks. This packages provides tools to visualise and analyse spatio-temporal point patterns on linear networks using first- and second-order summary statistics.

Depends R (>= 3.3.0), spatstat (>= 2.0-0)

Imports spatstat.univar, spatstat.geom, spatstat.random, spatstat.explore, spatstat.linnet, stats, graphics

Suggests plot3D, lattice

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Mehdi Moradi [aut, cre] (<https://orcid.org/0000-0003-3905-4498>), Ottmar Cronie [ctb], Jorge Mateu [ctb]

Repository CRAN

Date/Publication 2024-05-27 12:10:22 UTC

R topics documented:

as.lpp.stlpp 2
as.stlpp 3
as.tpp.stlpp 4
density.stlpp 5
density.tpp 6
densityVoronoi.stlpp 7
densityVoronoi.tpp 9

Eastbourne 10
easynet 11
Medellin 11
methods.stlpp 12
methods.tpp 13
rpoistlpp 14
rpoistpp 15
rthin.stlpp 16
STLg 17
STLginhom 18
STLK 19
STLKinhom 20
stlpp 22
tpp 23
unique.stlpp 24

Index **26**

`as.lpp.stlpp` *Methods for spatio-temporal point patterns on a linear network*

Description

This function projects an object of class `stlpp` into a linear network.

Usage

```
## S3 method for class 'stlpp'
as.lpp(x,...)
```

Arguments

`x`	an object of class `stlpp`
`...`	arguments passed to `as.lpp`

Details

This function projects the spatio-temporal point pattern x on the linear network L into L, giving its corresponding spatial point pattern.

Value

An object of class `lpp`.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

See Also

as.stlpp, lpp, as.lpp

Examples

```
data(easynet)
x <- runifpointOnLines(40, easynet)
t1 <- sample(1:10,40,replace=TRUE)
Y <- as.stlpp(x,t=t1,L=easynet)
as.lpp.stlpp(Y)
```

---

as.stlpp *Convert data to a spatio-temporal point pattern on a linear network*

Description

This function converts data to a spatio-temporal point pattern on a linear network.

Usage

```
as.stlpp(x,y,t,L)
```

Arguments

x, y, t	vectors of Cartesian coordinates and time occurrence. Alternatively, x can be of classes data.frame, ppp and lpp
L	linear network (object of class linnet)

Details

This function converts data to an object of class stlpp. Data can be of formats:

- x is of class class data.frame with three columns. Then columns are considered as Cartesian coordinates (i.e. x,y,t) and they will be converted to a spatio-temporal point pattern on the linear network L.
- x is a planar point pattern (class ppp). Then x will be converted to a spatio-temporal point pattern on the linear network L and with coresponding time vector t.
- x is a linear point pattern (class lpp). Then x will be converted to a spatio-temporal point pattern on the linear network L and with coresponding time vector t.
- x,y,t are vectors of same length where x,y are living on the corresponding network L.

Value

A spatio-temporal point pattern on a linear network. An object of class stlpp.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

See Also

stlpp

Examples

```
data(easynet)
x <- runifpointOnLines(40, easynet)
t1 <- sample(1:10,40,replace=TRUE)
Y <- as.stlpp(x,t=t1,L=easynet)

Z <- as.lpp.stlpp(Y)
t2 <- sample(1:10,40,replace=TRUE)
W <- as.stlpp(Z,t=t2)
```

---

as.tpp.stlpp *Convert data to a one-dimensional point pattern*

Description

This function converts an object of class stlpp to class tpp.

Usage

```
as.tpp.stlpp(X)
```

Arguments

X	an object of class stlpp

Details

This function projects the spatio-temporal point pattern X into its corresponding time domain T.

Value

An object of class tpp.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

See Also

as.stlpp, lpp, as.lpp

Examples

```
X <- rpoistlpp(10,1,2,easynet)
as.tpp.stlpp(X)
```

`density.stlpp` *Kernel estimation of intensity of spatio-temporal point patterns on a linear network*

Description

Kernel density estimation of a spatio-temporal point pattern on a linear network.

Usage

```
## S3 method for class 'stlpp'
density(x,lbw,tbw,at=c("points","pixels"),dimt=512,...)
```

Arguments

x	an object of class stlpp
lbw	network smoothing bandwidth
tbw	time smoothing bandwidth
at	string specifying whether to compute the intensity values at a grid of pixel locations and times (at="pixels") or only at the points of x (at="points"). default is to estimate the intensity at pixels
dimt	the number of equally spaced points at which the temporal density is to be estimated. see density
...	arguments passed to density.lpp

Details

Kernel smoothing is applied to the spatio-temporal point pattern x using methods in Moradi et al (2019). The function computes estimated intensities assuming first-order separability. Estimated intensity values of the marginal spatial point pattern on the linear network will be obtained using the fast kernel smoothing technique of Rakshit et al. (2019) and function densityQuick.lpp, whereas the estimated intensity values of the marginal temporal point pattern will be estimated using the function density.

If lbw and tbw are not given, then they will be selected using bw.nrd0 and bw.scott.iso respectively.

Value

If `at="points"`: a vector of intensity values at the data points of x. If `at="pixels"`: a list of images on linear network. Each image represents an estimated spatio-temporal intensity at a fixed time. Check the attributes for more accommodated outputs.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

`density`, `density.lpp`, `bw.nrd0`, `bw.scott.iso`

Examples

```
X <- rpoistlpp(.2,a=0,b=5,L=easynet)
density(X)
```

`density.tpp` *Kernel estimation of intensity of one-dimensional point patterns*

Description

Kernel estimation of intensity of one-dimensional point patterns.

Usage

```
## S3 method for class 'tpp'
density(x,tbw,at=c("points","pixels"),...)
```

Arguments

`x`	an object of class `tpp`
`tbw`	time smoothing bandwidth
`at`	string specifying whether to compute the intensity values at a grid of pixel locations (at="pixels") or only at the points of x (at="points"). default is to estimate the intensity at pixels
`...`	arguments passed to density

Details

A vector of intensity values.

Value

If `at="points"`: a vector of intensity values at the data points of x.

If `at="pixels"`: a vector of intensity values over a grid.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com> and Ottmar Cronie

References

Mateu, J., Moradi, M., & Cronie, O. (2019). Spatio-temporal point patterns on linear networks: Pseudo-separable intensity estimation. Spatial Statistics, 100400.

See Also

`density`, `bw.nrd0`

Examples

```
X <- tpp(sample(c(1:24),200,replace = TRUE))
plot(density(X))
```

---

densityVoronoi.stlpp *Intensity estimate of spatio-temporal point pattern using Voronoi-Dirichlet tessellation*

Description

This function performs adaptive intensity estimation for spatio-temporal point patterns on linear networks using Voronoi-Dirichlet tessellation.

Usage

```
## S3 method for class 'stlpp'
densityVoronoi(X, f = 1, nrep = 1, separable=FALSE, at=c("points","pixels"), dimt=128,...)
```

Arguments

X	an object of class stlpp
f	fraction (between 0 and 1 inclusive) of the data points that will be used to build a tessellation for the intensity estimate
nrep	number of independent repetitions of the randomised procedure
separable	logical. If FALSE, it then calculates a pseudo-separable estimate
at	string specifying whether to compute the intensity values at a grid of pixel locations and time (at="pixels") or only at the points of x (at="points"). default is to estimate the intensity at pixels
dimt	the number of equally spaced points at which the temporal density is to be estimated. see density
...	arguments passed to densityVoronoi.lpp

Details

This function computes intensity estimates for spatio-temporal point patterns on linear networks using Voronoi-Dirichlet tessellation. Both first-order separability and pseudo-separability assumptions are accommodated in the function.

If separable=TRUE, the estimated intensities will be a product of the estimated intensities on the network and those on time. Estimated intensity of the spatial component will be obtained using densityVoronoi.lpp, whereas estimated intensities of the temporal component will be obtained via densityVoronoi.tpp. If f=1, the function calculates the estimations based on the original Voronoi intensity estimator.

If separable=FALSE, the estimated intensities will be calculated based on a sub-sampling technique explained in Mateu et al. (2019). nrep sub-samples will be obtained from X based on a given retention probability f, the function densityVoronoi.stlpp, considering separable=TRUE and f=1, will be applied to each obtained sub-sample, and finally, the estimated intensities will be the sum of all obtained estimated intensities from all sub-samples divided by the (f * nrep).

Value

If at="points": a vector of intensity values at the data points of X.

If at="pixels": a list of images on a linear network. Each image represents an estimated spatio-temporal intensity at a fixed time.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com> and Ottmar Cronie

References

Mateu, J., Moradi, M., & Cronie, O. (2019). Spatio-temporal point patterns on linear networks: Pseudo-separable intensity estimation. Spatial Statistics, 100400.

See Also

densityVoronoi.lpp, density.stlpp

Examples

```
X <- rpoistlpp(.2,a=0,b=5,L=easynet)
densityVoronoi(X)
```

`densityVoronoi.tpp` *Intensity estimate of temporal point patterns using Voronoi-Dirichlet tessellation*

Description

This function performs adaptive intensity estimation for temporal point patterns using Voronoi-Dirichlet tessellation.

Usage

```
## S3 method for class 'tpp'
densityVoronoi(X, f = 1, nrep = 1, at=c("points","pixels"), dimt=128,...)
```

Arguments

`X`	an object of class `tpp`
`f`	fraction (between 0 and 1 inclusive) of the data points that will be used to build a tessellation for the intensity estimate
`nrep`	number of independent repetitions of the randomised procedure
`at`	string specifying whether to compute the intensity values at a grid of pixel locations and time (at="pixels") or only at the points of x (at="points"). default is to estimate the intensity at pixels
`dimt`	the number of equally spaced points at which the temporal density is to be estimated. see density
`...`	arguments passed to `densityVoronoi.lpp`

Details

This function computes intensity estimates for temporal point patterns using Voronoi-Dirichlet tessellation.

IF f<1, then nrep independent sub-samples of X are obtained using the function `rthin.stlpp`. Then for each of the obtained sub-samples, we calculate the Voronoi estimate. The final estimation is the sum of all obtained estimated intensities divided by (f*nrep).

Value

If `at="points"`: a vector of intensity values at the data points of X.

If `at="pixels"`: a vector of intensity values over a grid.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com> and Ottmar Cronie

References

Mateu, J., Moradi, M., & Cronie, O. (2019). Spatio-temporal point patterns on linear networks: Pseudo-separable intensity estimation. Spatial Statistics, 100400.

See Also

densityVoronoi.lpp, density.stlpp

Examples

```
X <- rpoistlpp(0.2,a=0,b=5,L=easynet)
Y <- as.tpp.stlpp(X)
densityVoronoi(Y)
```

Eastbourne *Eastbourne traffic accident data*

Description

This dataset represents the spatio-temporal locations of traffic accidents in the down-town of Eastbourne (UK) in the period of 2005-2010. The network was provided by "OS OpenData" at www.ordnancesurvey.co.uk and is usable under the terms of the OS OpenData license. The traffic locations were collected by the UK Department for Transport at www.data.gov.uk and obtained through kaggle at www.kaggle.com.

The dataset Eastbourne is an object of class stlpp.

Usage

```
data(Eastbourne)
```

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

Source

Usability: The network of Eastbourne was provided by OS OpenData and contains OS data © Crown copyright and database right (2018). The traffic accident locations in Eastbourne were collected by the UK Department for Transport and were provided by kaggle.

This data is a part of enitre data which is selected and converted to this format by Mehdi Moradi.

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

stlpp

Examples

```
data(Eastbourne)
plot(Eastbourne)
```

---

easynet *A simple linear network*

Description

A simple and not real network.

Usage

```
data(easynet)
```

Source

Created by Mehdi Moradi

Medellin *Medellin traffic accident data*

Description

This dataset represents the spatio-temporal locations of traffic accidents in an area near the pontifical bolivarian university in Medellin (Colombia) during 2016. The entire data were published in the OpenData portal of Medellin Town Hall at https://www.medellin.gov.co/geomedellin/index.hyg.

The dataset Medellin is an object of class stlpp.

Usage

```
data(Medellin)
```

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

Source

This data is a part of enitre data which is selected and converted to this format by Mehdi Moradi.

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

stlpp

Examples

```
data(Medellin)
plot(Medellin)
```

---

methods.stlpp *Methods for space-time point patterns on a linear network*

Description

Methods for space-time point patterns on a linear network.

Usage

```
## S3 method for class 'stlpp'
plot(x,xlab = xlab,...)
## S3 method for class 'stlppint'
plot(x,style=style,xlab=xlab,xlim=xlim,ylim=ylim,bar=TRUE,...)
## S3 method for class 'sumstlpp'
plot(x,style=c("level","contour","perspective"), theta = 35, phi = 10,
facets = FALSE, ticktype = "detailed", resfac = 5,xlab="r = distance",ylab="t = time",...)
## S3 method for class 'stlpp'
print(x,...)
## S3 method for class 'stlppint'
print(x,...)
## S3 method for class 'sumstlpp'
print(x,...)
## S3 method for class 'stlpp'
x[i]
## S3 method for class 'stlppint'
x[i]
## S3 method for class 'stlppint'
as.linim(X,...)
## S3 method for class 'stlppint'
as.tppint(x)
## S3 method for class 'sumstlpp'
as.data.frame(x,...)
```

Arguments

x,X	an object of classes stlpp, stlppint or sumstlpp
style	style of plot
theta,phi	see persp3D
facets,ticktype	see persp3D
resfac	see persp3D
xlab,ylab	the x,y label of the plot
xlim	giving the x limits for the plot
ylim	giving the y limits for the plot
bar	if TRUE, bar plot of rounded time occurances will be added to the density plot
i	numeric, logical, or an object of class stlpp
...	either ignore for as.linim, or gracphical arguments passed to plot/print

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

---

methods.tpp *Methods for one-dimensional point patterns*

Description

Methods for one-dimensional point patterns.

Usage

```
## S3 method for class 'tpp'
plot(x,xlab="time",ylab="",main = "cumulative number",...)
## S3 method for class 'tppint'
plot(x,xlab=xlab,xlim=xlim,line=2.5,main="NULL",...)
## S3 method for class 'tpp'
print(x,...)
## S3 method for class 'tppint'
print(x,...)
## S3 method for class 'tpp'
x[i]
## S3 method for class 'tppint'
x[i]
```

Arguments

x	an object of class tpp or tppint.
xlab,ylab	the x,y label of the plot.
main	overall title for the plot.
xlim	giving the x limits for the plot.
line	specifying a value for line overrides the default placement of y label, and places it this many lines outwards from the plot edge.
i	numeric, logical, or an object of class tpp
...	graphics parameters passed to plot/print function.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

Examples

```
X <- tpp(sample(c(1:24),200,replace = TRUE))
plot(X)
plot(density(X))
```

---

rpoistlpp *Simulating spatio-temporal Poisson point processes on a linear network*

Description

This function simulates realisations of a spatio-temporal Poisson point process on a linear network.

Usage

```
rpoistlpp(lambda,a,b,L,check=FALSE,lmax=NULL,nsim=1)
```

Arguments

lambda	intensity of the point process. it can be either a number, function of location and time, or an abject of class stlppint
a	lower bound of time period
b	upper bound of time period
L	a linear network
check	logical value indicating whether to check that all the (x,y) points lie inside the specified window. see ppp
lmax	upper bound for the values of labmda. this is optional
nsim	number of simulated patterns to generate

Details

This function generates realisations of a spatio-temporal poisson point process on a linear network based on an intensity function lambda and lower/upper bounds a and b.

Value

an object of class `stlpp` if nsim=1, otherwise a list of objects of class `stlpp`.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

`density.stlpp`

Examples

```
X <- rpoistlpp(0.2,a=0,b=5,L=easynet)
X
```

`rpoistpp` *Simulating one-dimensional Poisson point patterns*

Description

This function simulates realisations of an one-dimensional Poisson point process.

Usage

```
rpoistpp(lambda,a,b,check=FALSE,lmax=NULL,nsim=1)
```

Arguments

`lambda`	intensity of the point process. it can be either a number, a function of location and time, or an object of class `tppint`
`a`	lower bound of time period
`b`	upper bound of time period
`check`	logical value indicating whether to check that all the (x,y) points lie inside the specified time period.
`lmax`	upper bound for the values of `labmda`. this is optional
`nsim`	number of simulated patterns to generate

Details

This function generates realisations of a temporal poisson point process based on a given intensity function lambda and lower/upper bounds a and b.

Value

an object of class tpp if nsim=1, otherwise a list of objects of class tpp.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

rpoistlpp

Examples

```
f <- function(t){0.1*exp(t)}
X <- rpoistpp(f,a=1,b=10)
```

---

rthin.stlpp *Random thinning*

Description

This function applies independent random thinning to a spatio-temporal point pattern on a linear network.

Usage

```
## S3 method for class 'stlpp'
rthin(X, P = P, nsim = 1)
```

Arguments

X	a spatio-temporal point pattern of class stlpp
P	retention probability
nsim	number of simulated realisations to be generated

Details

See rthin.

Value

An object of the same kind as X if nsim=1, or a list of such objects if nsim > 1.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

stlpp, rthin

Examples

```
data(Medellin)
rthin(Medellin,P=.5)
```

STLg *Pair correlation function for spatio-temporal point processes on linear networks*

Description

This function computes the pair correlation function for spatio-temporal point patterns on linear networks.

Usage

```
STLg(X,r=NULL,t=NULL,nxy=10)
```

Arguments

X	a spatio-temporal point pattern of class stlpp
r	values of argument r where pair correlation function will be evaluated. optional
t	values of argument t where pair correlation function will be evaluated. optional
nxy	pixel array dimensions. optional

Details

This function calculates the pair correlation function for a homogeneous spatio-temporal point patterns on a linear network.

Value

An object of class `sumstlpp`.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

`pcf`, `STLK`

Examples

```
X <- rpoistlpp(.2,a=0,b=5,L=easynet)
g <- STLg(X)
plot(g)
```

`STLginhom` *Inhomogeneous pair correlation function for spatio-temporal point processes on linear networks*

Description

This function computes the inhomogeneous pair correlation function for spatio-temporal point patterns on linear networks.

Usage

```
STLginhom(X,lambda,normalize=FALSE,r=NULL,t=NULL,nxy=10)
```

Arguments

`X`	a spatio-temporal point pattern of class `stlpp`
`lambda`	values of estimated intensity at data points
`normalize`	normalization factor to be considered
`r`	values of argument r where pair correlation function will be evaluated. optional
`t`	values of argument t where pair correlation function will be evaluated. optional
`nxy`	pixel array dimensions. optional

Details

This function calculates the inhomogeneous pair correlation function for a spatio-temporal point patterns on a linear network.

Value

An object of class `sumstlpp`.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

`STLg`, `STLK`, `STLKinhom`

Examples

```
X <- rpoistlpp(.2,a=0,b=5,L=easynet)
d <- density(X,at="points")
g <- STLginhom(X,lambda=d,normalize=TRUE)
plot(g)
```

STLK *K-function for spatio-temporal point processes on linear networks*

Description

This function computes the K-function for spatio-temporal point patterns on linear networks.

Usage

```
STLK(X,r=NULL,t=NULL,nxy=10)
```

Arguments

`X`	a spatio-temporal point pattern of class `stlpp`
`r`	values of argument r where pair correlation function will be evaluated. optional
`t`	values of argument t where pair correlation function will be evaluated. optional
`nxy`	pixel array dimensions. optional

Details

This function calculates the K-function for a homogeneous spatio-temporal point patterns on a linear network.

Value

An object of class `sumstlpp`.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

`Kest`, `STLg`

Examples

```
X <- rpoistlpp(.2,a=0,b=5,L=easynet)
k <- STLK(X)
plot(k)
```

`STLKinhom` *Inhomogeneous K-function for spatio-temporal point processes on linear networks*

Description

This function computes the inhomogeneous K-function for spatio-temporal point patterns on linear networks.

Usage

```
STLKinhom(X,lambda=lambda,normalize=FALSE,r=NULL,t=NULL,nxy=10)
```

Arguments

X	a spatio-temporal point pattern of class stlpp
lambda	values of estimated intensity at data points
normalize	normalization factor to be considered
r	values of argument r where pair correlation function will be evaluated. optional
t	values of argument t where pair correlation function will be evaluated. optional
nxy	pixel array dimensions. optional

Details

This function calculates the inhomogeneous K-function for a spatio-temporal point patterns on a linear network.

Value

An object of class sumstlpp.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

STLg, STLK, STLginhom

Examples

```
X <- rpoistlpp(.2,a=0,b=5,L=easynet)
lambda <- density(X,at="points")
k <- STLKinhom(X,lambda=lambda,normalize=TRUE)
plot(k)
```

stlpp *Create spatio-temporal point pattern on linear network*

Description

Create an object of class stlpp representing a spatio-temporal point pattern on a linear network.

Usage

```
stlpp(X, L, T, ...)
```

Arguments

X	Locations of the points. a matrix or data frame of coordinates, or a point pattern object (of class "ppp") or other data acceptable to as.ppp or lpp
L	linear network (object of class linnet) on which the points lie
T	time occurrence of the points
...	ignored

Details

This function creates an object of class stlpp. For details about X see lpp. T represents the time occurrences of data points.

Value

An object of class stlpp.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

See Also

as.stlpp, lpp

Examples

```
data(easynet)
X <- rpoislpp(1,easynet)
t <- runif(npoints(X))
stlpp(X,T=t,L=easynet)
```

tpp *Create a temporal point pattern*

Description

Create an object of class tpp representing a one-dimensional point pattern.

Usage

```
tpp(X,a,b)
```

Arguments

X	an object of class numeric, integer or vector
a	lower band of the time domain. if not given by the user, it will be the minimum of X
b	upper bound of the time domain. if not given by the user, it will be the maximum of X

Details

Create a one-dimensional point pattern.

Value

An object of class tpp.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

See Also

stlpp

Examples

```
tpp(runif(10))
```

`unique.stlpp` *Extract unique points from a spatio-temporal point pattern on a linear network*

Description

This function extracts unique points from a spatio-temporal point pattern on a linear network.

Usage

```
## S3 method for class 'stlpp'
unique(x,...)
```

Arguments

`x`	a spatio-temporal point pattern of class `stlpp`
`...`	arguments for `unique`

Details

This function extracts unique points from a spatio-temporal point pattern on a linear network.

Value

A spatio-temporal point pattern on a linear network with no duplicated point.

Author(s)

Mehdi Moradi <m2.moradi@yahoo.com>

References

Moradi, M., & Mateu, J. (2020). First-and second-order characteristics of spatio-temporal point processes on linear networks. Journal of Computational and Graphical Statistics, 29(3), 432-443.

See Also

`unique`

Examples

```
X <-  rpoistlpp(0.1,0,5,L=easynet)
df <- as.data.frame(X)
df_dup <- df[sample(nrow(df), 20,replace = TRUE), ]
Y <- as.stlpp(df_dup,L=easynet)
npoints(Y)
npoints(unique(Y))
```

Index

[.stlpp *(methods.stlpp)*, 12
[.stlppint *(methods.stlpp)*, 12
[.tpp *(methods.tpp)*, 13
[.tppint *(methods.tpp)*, 13

as.data.frame.sumstlpp *(methods.stlpp)*, 12
as.linim, *13*
as.linim.stlppint *(methods.stlpp)*, 12
as.lpp, *2*, *3*, *5*
as.lpp.stlpp, 2
as.ppp, *22*
as.stlpp, *3*, 3, *5*, *22*
as.tpp.stlpp, 4
as.tppint.stlppint *(methods.stlpp)*, 12

bw.nrd0, *5–7*
bw.scott.iso, *5*, *6*

data.frame, *3*
density, *5–9*
density.lpp, *5*, *6*
density.stlpp, 5, *8*, *10*, *15*
density.tpp, 6
densityQuick.lpp, *5*
densityVoronoi.lpp, *8–10*
densityVoronoi.stlpp, 7, *8*
densityVoronoi.tpp, *8*, 9

Eastbourne, *10*, 10
easynet, 11

integer, *23*

Kest, *20*

linnet, *3*, *22*
lpp, *2*, *3*, *5*, *22*

Medellin, *11*, 11
methods.stlpp, 12
methods.tpp, 13

numeric, *23*

pcf, *18*
persp3D, *13*
plot, *13*
plot.stlpp *(methods.stlpp)*, 12
plot.stlppint *(methods.stlpp)*, 12
plot.sumstlpp *(methods.stlpp)*, 12
plot.tpp *(methods.tpp)*, 13
plot.tppint *(methods.tpp)*, 13
ppp, *3*, *14*
print, *13*
print.stlpp *(methods.stlpp)*, 12
print.stlppint *(methods.stlpp)*, 12
print.sumstlpp *(methods.stlpp)*, 12
print.tpp *(methods.tpp)*, 13
print.tppint *(methods.tpp)*, 13

rpoistlpp, 14, *16*
rpoistpp, 15
rthin, *17*
rthin.stlpp, *9*, 16

STLg, 17, *19–21*
STLginhom, 18, *21*
STLK, *18*, *19*, 19, *21*
STLKinhom, *19*, 20
stlpp, *2–5*, *8*, *10–13*, *15–19*, *21*, *22*, 22, *23*, *24*

tpp, *4*, *6*, *9*, *14*, *16*, *23*, 23

unique, *24*
unique.stlpp, 24

vector, *23*