Package ‘stopp’

June 4, 2024

Title  Spatio-Temporal Point Pattern Methods, Model Fitting, Diagnostics, Simulation, Local Tests

Version 0.2.4

Date 2024-06-03

Type Package


License GPL (>= 2)

Encoding UTF-8

Depends R (>= 4.0.0)

Imports stats, graphics, KernSmooth, MASS, fields, optimx, plot3D, sparr, spatstat.explore, spatstat.geom, spatstat.lilnet, spatstat.random, splancs, spatstat.model, spatstat.utils, stlnpp, stpp, mgcv, spatstat.univar

LazyData true

Author Nicoletta D’Angelo [aut, cre] (<https://orcid.org/0000-0002-8878-5986>), Giada Adelfio [aut] (<https://orcid.org/0000-0002-3194-4296>)

Maintainer Nicoletta D’Angelo <nicoletta.dangelo@unipa.it>

RoxygenNote 7.3.1

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2024-06-04 09:47:54 UTC
R topics documented:

- stopp-package ......................................................... 3
- chicagonet ............................................................ 4
- globaldiag ............................................................ 5
- greececatalog ....................................................... 6
- infl ................................................................. 8
- localdiag ........................................................... 9
- localplot ........................................................... 11
- localSTLginhom ..................................................... 12
- localSTLKinhom .................................................... 13
- localsummary ....................................................... 15
- localtest ........................................................... 16
- locstppm ............................................................ 18
- plot.globaldiag ..................................................... 20
- plot.lista ........................................................... 22
- plot.localdiag ........................................................ 23
- plot.localtest ........................................................ 24
- plot.locstppm ......................................................... 25
- plot.sepstppm ....................................................... 27
- plot.sepstppm ....................................................... 27
- plot.stcov .......................................................... 29
- plot.stlgcppm ....................................................... 29
- plot.stlp ............................................................ 31
- plot.stlpn .......................................................... 32
- plot.stp ............................................................. 33
- plot.stppm .......................................................... 35
- print.globaldiag .................................................... 36
- print.lista .......................................................... 38
- print.localdiag ...................................................... 38
- print.localtest ........................................................ 40
- print.locstppm ....................................................... 41
- print.sepstppm ....................................................... 42
- print.sepstppm ....................................................... 43
- print.stcov .......................................................... 44
- print.stlgcppm ....................................................... 44
- print.stlp ............................................................ 45
- print.stlpn .......................................................... 46
- print.stp ............................................................. 47
- print.stpm .......................................................... 48
- print.stppm .......................................................... 48
- rETASlp .............................................................. 49
- rETASp .............................................................. 51
- rstlpp .............................................................. 53
- rstpp .............................................................. 55
- sepstppm ............................................................ 56
- sepstpppm ........................................................... 57
stopp-package

Spatio-Temporal Point Pattern Methods, Model Fitting, Diagnostics, Simulation, Local Tests

Description

Toolbox for different kinds of spatio-temporal analyses to be performed on observed point patterns, following the growing stream of literature on point process theory. This R package implements functions to perform different kinds of analyses on point processes, proposed in the papers: Siino, Adelfio, and Mateu (2018), Siino et al. (2018), Adelfio et al. (2020), D’Angelo, Adelfio, and Mateu (2021), D’Angelo, Adelfio, and Mateu (2022), and D’Angelo, Adelfio, and Mateu (2023). The main topics include modeling, statistical inference, and simulation issues on spatio-temporal point processes on Euclidean space and linear networks.

Author(s)

Nicoletta D’Angelo [aut,cre] nicoletta.dangelo@unipa.it, Giada Adelfio [aut]

References


---

**chicagonet**  
Rescaled roads of Chicago (Illinois, USA)

**Description**

A linear network of class `linnet` of the roads of Chicago (Illinois, USA) close to the University of Chicago. The window has been rescaled to be enclosed in a unit square.

**Usage**

data(chicagonet)

**Format**

A linear network of class `linnet`

**Author(s)**

Nicoletta D’Angelo

**References**


**Examples**

data(chicagonet)
Global diagnostics of a spatio-temporal point process first-order intensity

Description
This function performs global diagnostics of a model fitted for the first-order intensity of a spatio-temporal point pattern, by returning the inhomogeneous K-function weighted by the provided intensity to diagnose, its theoretical value, and their difference.

Usage
globaldiag(x, intensity)

Arguments
x A stp object
intensity A vector of intensity values, of the same length as the number of point in x

Details
If applied to a stp object, it resorts to the spatio-temporal inhomogeneous K-function (Gabriel and Diggle, 2009) documented by the function STIKhat of the stpp package (Gabriel et al, 2013).

If applied to a stlp object, it uses the spatio-temporal inhomogeneous K-function on a linear network (Moradi and Mateu, 2020) documented by the function STLKinhom of the stlnpp package (Moradi et al., 2020).

Value
A list of class globaldiag, containing
x The observed point pattern
dist The spatial ranges of the K-function	imes The temporal ranges of the K-function
est The estimated K-function weighted by the intensity function in input
theo The theoretical K-function
diffK The difference between the estimated and the theoretical K-functions
squared.diff The sum of the squared differences between the estimated and the theoretical K-functions

Author(s)
Nicoletta D’Angelo and Giada Adelfio
References


See Also

plot.globaldiag, print.globaldiag, summary.globaldiag

Examples

```r
set.seed(2)
inl <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(.3, 6))

mod1 <- stppm(inl, formula = ~ 1)
mod2 <- stppm(inl, formula = ~ x)

g1 <- globaldiag(inl, mod1$l)
g2 <- globaldiag(inl, mod2$l)
```

Description

A dataset in stp format containing the catalog of Greek earthquakes of magnitude at least 4.0 from year 2005 to year 2014. Data come from the Hellenic Unified Seismic Network (H.U.S.N.).
Usage

data(greececatalog)

Format

A stp object for a spatio-temporal point pattern with 1111 points

Details

The variables are as follows:

- \texttt{x}. longitude, ranging from 20.02 to 27.98
- \texttt{y}. latitude, ranging from 33.75 to 40.45
- \texttt{t}. time, ranging from 38354, 42000

Author(s)

Nicoletta D’Angelo

References


Examples

data(greececatalog)

plot(greececatalog)
Display outlying LISTA functions

Description

This function works on the objects of class `localdiag`, as returned by `localdiag`, plotting the identified 'outlying' LISTA functions. These correspond to the influential points in the fitting of the model provided by `localdiag`

Usage

```r
infl(x, id = NULL)
```

Arguments

- `x` An object of class `localdiag`
- `id` The id of the LISTA to display. Default is set to the ids identified and stored in the `localdiag` object

Author(s)

Nicoletta D’Angelo and Giada Adelfio

References


See Also

`localdiag`, `plot.localdiag`, `print.localdiag`, `summary.localdiag`

Examples

```r
set.seed(2)
in <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(.3, 6))

mod1 <- stppm(in, formula = ~ 1)
resmod1 <- localdiag(in, mod1$l, p = .9)
infl(resmod1)
```
Description

This function performs local diagnostics of a model fitted for the first-order intensity of a spatio-temporal point pattern, returning the points identified as outlying following the diagnostics procedure on individual points of an observed point pattern, as introduced in Adelfio et al. (2020), and applied in D’Angelo et al. (2022) for the linear network case.

The points resulting from the local diagnostic procedure provided by this function can be inspected via the plot, print, summary, and infl functions.

Usage

localdiag(x, intensity, p = 0.95)

Arguments

- **x**: Either a stp or a stlp object
- **intensity**: A vector of intensity values, of the same length as the number of point in x
- **p**: The percentile to consider as threshold for the outlying points. Default to 0.95.

Details

This function performs local diagnostics of a model fitted for the first-order intensity of a spatio-temporal point pattern, by means of the local spatio-temporal inhomogeneous K-function (Adelfio et al, 2020) documented by the function KLISTAhat of the stpp package (Gabriel et al, 2013).

The function can also perform local diagnostics of a model fitted for the first-order intensity of an spatio-temporal point pattern on a linear network, by means of the local spatio-temporal inhomogeneous K-function on linear networks (D’Angelo et al, 2021) documented by the function localSTLKinhom.

In both cases, it returns the points identified as outlying following the diagnostics procedure on individual points of an observed point pattern, as introduced in Adelfio et al. (2020), and applied in D’Angelo et al. (2022) for the linear network case.

This function computes discrepancies by means of the $\chi^2_i$ values, obtained following the expression

$$
\chi^2_i = \int_L \int_T \left( \frac{(\hat{K}_i^*(r, h) - \mathbb{E}[\hat{K}_i^*(r, h)])}{\mathbb{E}[\hat{K}_i^*(r, h)]} \right)^2 \, dh \, dr,
$$

one for each point in the point pattern.

Note that the Euclidean procedure is implemented by the local K-functions of Adelfio et al. (2020), documented in KLISTAhat of the stpp package (Gabriel et al, 2013). The network case uses the local K-functions on networks (D’Angelo et al., 2021), documented in localSTLKinhom.
Value

A list object of class `localdiag`, containing

- x The stp object provided as input
- listas The LISTA functions, in a list object
- ids The ids of the points identified as outlying
- x2 A vector with the individual contributions to the Chi-squared statistics, normalized
- p The percentile considered

Author(s)

Nicoletta D’Angelo and Giada Adelfio

References


See Also

`infl`, `plot.localdiag`, `print.localdiag`, `summary.localdiag`, `globaldiag`

Examples

```r
set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
             par = c(.3, 6))
mod1 <- stppm(inh, formula = ~ 1)
resmod1 <- localdiag(inh, mod1$l, p = .9)
```
Function: \texttt{localplot}

\textit{Plot the coefficients of a fitted local spatio-temporal Poisson process or local LGCP model}

**Description**

The function plots the local estimates of a fitted local spatio-temporal Poisson process or local LGCP model.

**Usage**

\begin{verbatim}
localplot(x, par = TRUE)
\end{verbatim}

**Arguments**

- \texttt{x}: An object of class \texttt{locstppm} or \texttt{stlgcppm}.
- \texttt{par}: Default to \texttt{TRUE}.

**Author(s)**

Nicoletta D’Angelo

**References**


**See Also**

\texttt{locstppm}, \texttt{stlgcppm}

**Examples**

\begin{verbatim}
# Local spatio-temporal Poisson process model
set.seed(2)
inhibit <- rstpp(lambda = function(x, y, t, a) \{exp(a[1] + a[2]*x)},
par = c(0.005, 5))
inhibit_local <- locstppm(inhibit, formula = ~ x)
localplot(inhibit_local)

# Local LGCP

catsub <- stp(greececatalog$df[1:200,])
lgcp_loc <- stlgcppm(catsub, formula = ~ x, first = "local")
\end{verbatim}
localSTLginhom

Local inhomogeneous Spatio-temporal pair correlation functions on a linear network

Description
The functions localSTLKinhom and localSTLginhom implement the inhomogeneous LISTA functions proposed in D’Angelo et al. (2022).

Usage
localSTLginhom(x, lambda, normalize = FALSE, r = NULL, t = NULL, nxy = 10)

Arguments
- **x**: A realisation of a spatio-temporal point processes on a linear network in stlp format.
- **lambda**: values of estimated intensity.
- **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
The homogeneous K-function and pair correlation functions, in D’Angelo et al. (2021), can be obtained easily with localSTLKinhom and localSTLginhom, by imputing a lambda vector of constant intensity values, the same for each point.

Value
A list of class lista. The objects are of class sumstlpp (Moradi and Mateu, 2020).

Author(s)
Nicoletta D’Angelo


References


See Also

localSTLginhom, STLKinhom, STLginhom

Examples

```r
set.seed(2)
df_net <- data.frame(x = runif(25, 0, 0.85), y = runif(25, 0, 0.85), t = runif(25))
stlp1 <- stp(df_net, L = chicagonet)
lambda <- rep(diff(range(stlp1$df$x)) * diff(range(stlp1$df$y)) * diff(range(stlp1$df$t)) / spatstat.geom::volume(stlp1$L), nrow(stlp1$df))

g <- localSTLginhom(stlp1, lambda = lambda, normalize = TRUE)
```

---

**localSTLKinhom**

*Local inhomogeneous Spatio-temporal K-functions on a linear network*

Description

The functions `localSTLKinhom` and `localSTLginhom` implement the inhomogeneous LISTA functions proposed in D’Angelo et al. (2022).

Usage

```r
localSTLKinhom(
  x,
  lambda = lambda,
  normalize = FALSE,
  r = NULL,
  t = NULL,
  nxy = 10
)
```
Arguments

- **x**: A realisation of a spatio-temporal point processes on a linear network in `stlp` format.
- **lambda**: values of estimated intensity.
- **normalize**: normalization factor to be considered.
- **r**: values of argument r where K-function will be evaluated. optional.
- **t**: values of argument t where K-function will be evaluated. optional.
- **nxy**: pixel array dimensions. optional.

Details

The homogeneous K-function and pair correlation functions, in D’Angelo et al. (2021), can be obtained easily with `localSTLKinhom` and `localSTLginhom`, by imputing a lambda vector of constant intensity values, the same for each point.

Value

A list of class `lista`. The objects are of class `sumstlpp` (Moradi and Mateu, 2020).

Author(s)

Nicoletta D’Angelo

References


See Also

`localSTLginhom`, `STLKinhom`, `STLginhom`

Examples

```r
set.seed(2)
df_net <- data.frame(x = runif(25, 0, 0.85), y = runif(25, 0, 0.85), t = runif(25))
stlp1 <- stp(df_net, L = chicagonet)
lambda <- rep(diff(range(stlp1$df$x)) * diff(range(stlp1$df$y)) * diff(range(stlp1$df$t)) / spatstat.geom::volume(stlp1$L), nrow(stlp1$df))

k <- localSTLKinhom(stlp1, lambda = lambda, normalize = TRUE)
```
Summary plots of the fitted coefficient of a local spatio-temporal Poisson process or a local LGCP model

Description
The function breaks up the contribution of the local estimates to the fitted intensity, by plotting the overall intensity and the density kernel smoothing of some artificial intensities, obtained by imputing the quartiles of the local parameters’ distributions.

Usage
localsummary(
  x, 
  scaler = c("silverman", "IQR", "sd", "var"), 
  do.points = TRUE, 
  print.bw = FALSE, 
  zap = 1e-05, 
  par = TRUE 
)

Arguments

x An object of class locstppm or stlgcppm
scaler Optional. Controls the value for a scalar representation of the spatial scale of the data. Either a character string, "silverman" (default), "IQR", "sd", or "var"; or positive numeric value(s). See OS.
do.points Add points to plot
print.bw It prints the estimated oversmoothing (OS) bandwidth selector
zap Noise threshold factor (default to 0.00001). A numerical value greater than or equal to 1. If the range of pixel values is less than zap * .Machine$double.eps, the image will be treated as constant. This avoids displaying images which should be constant but contain small numerical errors.
par Default to TRUE.

Author(s)
Nicoletta D’Angelo and Giada Adelfio

References

See Also

locstppm, stlgcppm

Examples

# Local spatio-temporal Poisson process model

```r
set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
par = c(0.005, 5))
inh_local <- locstppm(inh, formula = ~ x)
localsummary(inh_local)
```

# Local LGCP

```r
catsub <- stp(greececatalog$df[1:200, ])
lgcp_loc <- stlgcppm(catsub, formula = ~ x, first = "local")
localsummary(lgcp_loc)
```

localtest

Test of local structure for spatio-temporal point processes

Description

This function performs the permutation test of the local structure for spatio-temporal point pattern data, proposed in Siino et al. (2018), as well as for spatio-temporal point pattern data occurring on the same linear network, following D’Angelo et al. (2021).

Usage

```r
localtest(X, Z, method = c("K", "g"), k, alpha = 0.05, verbose = TRUE)
```

Arguments

- **X**: Background spatio-temporal point pattern. Usually, the most clustered between X and Z. Must be either a stp or stlp object.
- **Z**: Other spatio-temporal point pattern. Must also be of the same class as X.
localtest

- **method**: Character string indicating which version of LISTA function to use: either "K" or "g". If "K", the local spatio-temporal K-function is used to run the test. If "g", the local spatio-temporal pair correlation function is used.
- **k**: Number of permutations
- **alpha**: Significance level
- **verbose**: If TRUE (default) the progress of the test is printed

**Details**

The test detects local differences between \( x \) and \( z \) occurring on the same space-time region.

The test ends providing a vector \( p \) of \( p \)-values, one for each point in \( x \).

If the test is performed for spatio-temporal point patterns as in Siino et al. (2018), that is, on an object of class \( \text{stp} \), the LISTA functions \( \hat{L}^{(i)} \) employed are the local functions of Adelfio et al. (2020), documented in \( \text{KLISTAhat} \) and \( \text{LISTAhat} \) of the \text{stpp} package (Gabriel et al, 2013).

If the function is applied to a \( \text{stlp} \) object, that is, on two spatio-temporal point patterns observed on the same linear network \( L \), the LISTA function \( \hat{L}^{(i)} \) used are the ones proposed in D’Angelo et al. (2021), documented in \( \text{localSTLKinhom} \) and \( \text{localSTLginhom} \).

Details on the performance of the test are found in Siino et al. (2018) and D’Angelo et al. (2021), for Euclidean and network spaces, respectively.

**Value**

A list of class \text{localtest}, containing

- \( p \): A vector of \( p \)-values, one for each of the points in \( X \)
- \( X \): The background spatio-temporal point pattern given in input
- \( Z \): The alternative spatio-temporal point pattern given in input
- \( \text{alpha} \): The threshold given in input
- \( X\text{sig} \): A \text{stp} object storing the resulting significant points
- \( X\text{nosig} \): A \text{stp} object storing the resulting non-significant points
- \( \text{id} \): The ids of the resulting significant points

**Author(s)**

Nicoletta D’Angelo and Marianna Siino

**References**


See Also

print.localtest, summary.localtest, plot.localtest

Examples

```r
set.seed(2)
X <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
        par = c(.005, 5))
Z <- rstpp(lambda = 30)

test <- localtest(X, Z, method = "K", k = 3)
```

locstppm

Fit a local Poisson process model to a spatio-temporal point pattern

Description

This function fits a Poisson process model to an observed spatio-temporal point pattern stored in a stp object, that is, a Poisson model with a set of parameters \( \theta_i \) for each point \( i \).

Usage

```r
locstppm(
  X,
  formula,
  verbose = TRUE,
  mult = 4,
  seed = NULL,
  hs = c("global", "local"),
  npx0 = 10,
  npt0 = 10
)
```
**Arguments**

- **X**: A `stp` object
- **formula**: An object of class "formula": a symbolic description of the model to be fitted. The current version only supports formulas depending on the spatial and temporal coordinates: x, y, t.
- **verbose**: Default to TRUE
- **mult**: The multiplicand of the number of data points, for setting the number of dummy points to generate for the quadrature scheme
- **seed**: The seed used for the simulation of the dummy points. Default to NULL.
- **hs**: Character string indicating whether to select fixed or variable bandwidths for the kernel weights to be used in the log-likelihood. In any of those cases, the well-supported rule-of-thumb for choosing the bandwidth of a Gaussian kernel density estimator is employed. If `hs = "global"` (default), a fixed bandwidth is selected. If `hs = "local"`, an individual bandwidth is selected for each point in the pattern X.
- **npx0**: Number of lags for the space grid period for variable bandwidths kernel
- **npt0**: Number of lags for the time period for variable bandwidths kernel

**Details**

We assume that the template model is a Poisson process, with a parametric intensity or rate function \( \lambda(u, t; \theta_i) \) with space and time locations \( u \in W, t \in T \) and parameters \( \theta_i \in \Theta \).

Estimation is performed through the fitting of a `glm` using a localized version of the quadrature scheme by Berman and Turner (1992), firstly introduced in the purely spatial context by Baddeley (2017), and in the spatio-temporal framework by D’Angelo et al. (2023).

**Value**

An object of class `locstppm`. A list of

- **IntCoefs**: The fitted global coefficients
- **IntCoefs_local**: The fitted local coefficients
- **X**: The `stp` object provided as input
- **nX**: The number of points in X
- **I**: Vector indicating which points are dummy or data
- **y_resp**: The response variable of the model fitted to the quadrature scheme
- **formula**: The formula provided as input
- **l**: Fitted intensity through the global parameters
- **l_local**: Fitted intensity through the local parameters
- **mod_global**: The `glm` object of the model fitted to the quadrature scheme
- **newdata**: The data used to fit the model, without the dummy points
- **time**: Time elapsed to fit the model, in minutes
Author(s)
Nicoletta D’Angelo

References

See Also
stppm

Examples
set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
par = c(0.005, 5))
inh_local <- locstpp(inh, formula = ~ x)

plot.globaldiag

Plot of the global diagnostics of a spatio-temporal point process first-order intensity

Description
This function performs global diagnostics of a model fitted for the first-order intensity of a spatio-temporal point pattern, by returning the plots of the inhomogeneous K-function weighted by the provided intensity to diagnose, its theoretical value, and their difference.

Usage
## S3 method for class 'globaldiag'
plot(x, samescale = TRUE, ...)

Arguments
x A globaldiag object
samescale Logical value. It indicates whether to plot the observed and the theoretical K-function in the same or different scale. Default to TRUE.
... additional unused argument
plot.globaldiag

Value

It plots three panels: the observed K-function, as returned by \texttt{STLKinhom}; the theoretical one; their difference. The function also prints the sum of squared differences between the observed and theoretical K-function on the console.

Author(s)

Nicoletta D’Angelo

References


See Also

globaldiag, print.globaldiag, summary.globaldiag

Examples

```r
set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
par = c(.3, 6))

mod1 <- stppm(inh, formula = ~ 1)
mod2 <- stppm(inh, formula = ~ x)

g1 <- globaldiag(inh, mod1$l)
g2 <- globaldiag(inh, mod2$l)

plot(g1)
plot(g2)
```
plot.lista

Display LISTA functions

Description

This function works on the objects of class lista, as returned by localSTLKinhom or localSTLginhom, plotting the specified LISTA functions.

Usage

## S3 method for class 'lista'
plot(x, id, ...)

Arguments

- **x**: An object of class lista
- **id**: The id of the LISTA to display
- **...**: additional unused argument

Author(s)

Nicoletta D’Angelo

References


See Also

localSTLKinhom, localSTLginhom

Examples

```r
set.seed(2)
df_net <- data.frame(x = runif(25, 0, 0.85), y = runif(25, 0, 0.85), t = runif(25))
stlp1 <- stp(df_net, L = chicagonet)
lambda <- rep(diff(range(stlp1$df$x)) * diff(range(stlp1$df$y))
          * diff(range(stlp1$df$t)) / spatstat.geom::volume(stlp1$L),
nrow(stlp1$df))
k <- localSTLKinhom(stlp1, lambda = lambda, normalize = TRUE)
```
plot.localdiag

plot(A, id = 1:9)

---

**Description**

This function plots the result of the local diagnostics performed with `localdiag` on either a `stp` or `stlp` object. It highlights the points of the analysed spatio-temporal point pattern $X$ which are identified as outlying by the previously performed local diagnostics; the remaining points of $X$ are also represented.

It also shows the underlying linear network, if the local diagnostics has been applied to point patterns occurring on the same linear network, that is, if `localdiag` has been applied to a `stlp` object.

**Usage**

```r
## S3 method for class 'localdiag'
plot(x, marg = TRUE, col = "grey", col2 = "red", cols = "lightgrey", ...)
```

**Arguments**

- `x` A `localdiag` object
- `marg` Default to `TRUE`. If `marg = F`, only the spatio-temporal point pattern is plotted
- `col` Color of the outlying points
- `col2` Color of the network (if applicable)
- `cols` Color of the non-outlying points
- `...` additional unused argument

**Author(s)**

Nicoletta D’Angelo and Giada Adelfio

**References**


See Also

\texttt{infl}, \texttt{print.localdiag}, \texttt{summary.localdiag}

Examples

```r
set.seed(2)
inl <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
            par = c(.3, 6))

mod1 <- stppm(inh, formula = ~ 1)

resmod1 <- localdiag(inh, mod1$l, p = .9)

plot(resmod1)
plot(resmod1, marg = FALSE)
```

---

\textbf{Description}

This function plots the result of the local permutation test performed with \texttt{localtest} on either a \texttt{stp} or \texttt{stlp} object. It highlights the points of the background pattern $X$, which exhibit local differences in the second-order structure with respect to $Z$, according to the previously performed test. The remaining points of $X$ are also represented.

It also shows the underlying linear network, if the local test has been applied to point patterns occurring on the same linear network, that is, if \texttt{localtest} has been applied to a \texttt{stlp} object.

\textbf{Usage}

```r
## S3 method for class 'localtest'
plot(x, col = "grey", cols = "lightgrey", col2 = "red", ...)
```

\textbf{Arguments}

\begin{itemize}
\item \texttt{x} \hspace{1cm} An object of class \texttt{localtest}
\item \texttt{col} \hspace{1cm} Color of the significant points
\item \texttt{cols} \hspace{1cm} Color of the linear network. If applicable.
\item \texttt{col2} \hspace{1cm} Color of the non-significant points
\item \ldots \hspace{1cm} additional unused argument
\end{itemize}

\textbf{Author(s)}

Nicoletta D’Angelo
References


See Also

localtest, print.localtest, summary.localtest

Examples

```r
set.seed(2)
X <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
           par = c(.005, 5))
Z <- rstpp(lambda = 30)

test <- localtest(X, Z, method = "K", k = 3)

plot(test)
```

---

**plot.locstppm**

*Plot of the fitted intensity of a local spatio-temporal Poisson process model*

Description

The function plots the local fitted intensity, displayed both in space and in space and time.

Usage

```r
## S3 method for class 'locstppm'
plot(
  x,
  scaler = c("silverman", "IQR", "sd", "var"),
  do.points = TRUE,
  print.bw = FALSE,
  zap = 1e-05,
  par = TRUE,
  ...
)
```
Arguments

- `x`: An object of class `locstppm`
- `scaler`: Optional. Controls the value for a scalar representation of the spatial scale of the data. Either a character string, "silverman" (default), "IQR", "sd", or "var"; or positive numeric value(s). See OS.
- `do.points`: Add points to plot
- `print.bw`: It prints the estimated oversmoothing (OS) bandwidth selector
- `zap`: Noise threshold factor (default to 0.00001). A numerical value greater than or equal to 1. If the range of pixel values is less than `zap * .Machine$double.eps`, the image will be treated as constant. This avoids displaying images which should be constant but contain small numerical errors.
- `par`: Default to TRUE.
  - Additional unused argument

Author(s)

Nicoletta D’Angelo

References


See Also

`locstppm`, `print.locstppm`, `summary.locstppm`

Examples

```r
set.seed(2)
ish <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
          par = c(0.005, 5))
inh_local <- locstppm(inh, formula = ~ x)
plot(inh_local)
```
plot.seplppm

Plot of the fitted intensity of a separable spatio-temporal Poisson model

Description

The function plots the fitted intensity, displayed both in space and in space and time.

Usage

```r
## S3 method for class 'seplppm'
plot(x, do.points = TRUE, par = TRUE, ...)
```

Arguments

- `x`: An object of class `seplppm`
- `do.points`: Add points to plot
- `par`: Default to `TRUE`. If `par=FALSE`, the user is asked for input, before a new figure is drawn.
- `...`: additional unused argument

Author(s)

Nicoletta D’Angelo

Examples

```r
crimesub <- stpm(valenciacrimes$df[101:200, ],
  names = colnames(valenciacrimes$df)[-c(1:3)],
  L = valencianet)

mod1 <- seplppm(crimesub, spaceformula = ~x,
  timeformula = ~ day)

plot(mod1)
```
Usage

```r
## S3 method for class 'sepstppm'
plot(
  x,
  scaler = c("silverman", "IQR", "sd", "var"),
  do.points = TRUE,
  print.bw = FALSE,
  zap = 1e-05,
  par = TRUE,
  sig = NULL,
  ...
)
```

Arguments

- `x`: An object of class `sepstppm`
- `scaler`: Optional. Controls the value for a scalar representation of the spatial scale of the data. Either a character string, "silverman" (default), "IQR", "sd", or "var"; or positive numeric value(s). See OS.
- `do.points`: Add points to plot
- `print.bw`: It prints the estimated oversmoothing (OS) bandwidth selector
- `zap`: Noise threshold factor (default to 0.00001). A numerical value greater than or equal to 1. If the range of pixel values is less than `zap * .Machine$double.eps`, the image will be treated as constant. This avoids displaying images which should be constant but contain small numerical errors.
- `par`: Default to TRUE. If par=FALSE, the user is asked for input, before a new figure is drawn.
- `sig`: Smoothing bandwidth for spatial representation
- `...`: additional unused argument

Author(s)

Nicoletta D’Angelo

Examples

```r
crimesub <- stpm(valenciacrimes$df[1:100, ],
                  names = colnames(valenciacrimes$df)[-c(1:3)])

mod1 <- sepstppm(crimesub, spaceformula = ~x ,
                 timeformula = ~ day)

plot(mod1)
```
**plot.stcov**  
*Plot a stcov object*

**Description**

This function plots the covariate stored in the *stcov* object given in input, in a three panel plot representing the 3Dplot of the coordinates, and the covariate values.

**Usage**

```r
## S3 method for class 'stcov'
plot(x, ...
```

**Arguments**

- **x**  
  An object of class stcov
- **...**  
  additional unused argument

**Author(s)**

Nicoletta D’Angelo

**See Also**

- stcov

**Examples**

```r
set.seed(2)
df <- data.frame(runif(100), runif(100), runif(100), rpois(100, 15))
cov <- stcov(df, interp = FALSE)
plot(cov)
```

---

**plot.stlgcppm**  
*Plot of the fitted intensity of a LGCP model*

**Description**

The function plots the fitted intensity, displayed both in space and in space and time. In the case of local covariance parameters, the function returns the mean of the random intensity, displayed both in space and in space and time.
Usage

```r
## S3 method for class 'stlgcppm'
plot(
x,
scaler = c("silverman", "IQR", "sd", "var"),
do.points = TRUE,
print.bw = FALSE,
zap = 1e-05,
par = TRUE,
...
)
```

Arguments

- **x** An object of class stlgcppm
- **scaler** Optional. Controls the value for a scalar representation of the spatial scale of the data. Either a character string, "silverman" (default), "IQR", "sd", or "var"; or positive numeric value(s). See OS.
- **do.points** Add points to plot
- **print.bw** It prints the estimated oversmoothing (OS) bandwidth selector
- **zap** Noise threshold factor (default to 0.00001). A numerical value greater than or equal to 1. If the range of pixel values is less than zap * \$.Machine$double.eps, the image will be treated as constant. This avoids displaying images which should be constant but contain small numerical errors.
- **par** Default to TRUE.
- **...** additional unused argument

Author(s)

Nicoletta D’Angelo and Giada Adelfio

References


See Also

stlgcppm, print.stlgcppm, summary.stlgcppm, localsummary, localplot
plot.stlp

Examples

catsub <- stp(greececatalog$df[1:200, ])
lgcp_loc <- stlgcppm(catsub, formula = - x, first = "local")
plot(lgcp_loc)

plot.stlp

Plot a stlp object

Description

This function plots the point pattern on a linear network stored in the stlp object given in input, in a three panel plot representing the plot3D of the coordinates, and the marginal spatial and temporal coordinates.

Usage

## S3 method for class 'stlp'
plot(x, tcum = TRUE, marg = TRUE, col = 1, cols = "grey", ...)

Arguments

x

An object of class stp

tcum

If TRUE (default option), the temporal point pattern is displayed cumulatively. A barplot is automatically plotted if there are repeated counts (typically with discrete times).

marg

Default to TRUE. If FALSE, only the spatio-temporal point pattern is plotted.

col

The color of the points. Default to "black"

cols

The color of the linear network. Default to "grey"

...

additional unused argument

Author(s)

Nicoletta D’Angelo

See Also

stp, summary.stlp, print.stlp
Examples

```r
set.seed(2)
df_net <- data.frame(cbind(runif(100, 0, 0.85), runif(100, 0, 0.85), runif(100)))
stlp1 <- stp(df_net, L = chicagonet)
plot(stlp1)
```

---

**plot.stlpm**

*Plot a stlpm object*

Description

This function plots the covariate stored in the stcov object given in input, in a three panel plot representing the 3Dplot of the coordinates, and the mark values.

Usage

```r
## S3 method for class 'stlpm'
plot(x, ...)
```

Arguments

- `x` An object of class `stpm`
- `...` additional unused argument

Author(s)

Nicoletta D’Angelo

See Also

`stppm`

Examples

```r
set.seed(2)
df <- data.frame(x = runif(100, 0, 0.8), y = runif(100, 0, 0.8), t = runif(100), m = rpois(100, 15))
stlpm1 <- stpm(df, L = chicagonet)
plot(stlpm1)
```
plot.stp

Plot a stp object

Description
This function plots the point pattern stored in the stp object given in input, in a three panel plot representing the 3D plot of the coordinates, and the marginal spatial and temporal coordinates.

Usage
## S3 method for class 'stp'
plot(x, tcum = TRUE, marg = TRUE, col = 1, ...)

Arguments
- **x**: An object of class stp
- **tcum**: If TRUE (default option), the temporal point pattern is displayed cumulatively. A barplot is automatically plotted if there are repeated counts (typically with discrete times).
- **marg**: Default to TRUE. If FALSE, only the spatio-temporal point pattern is plotted.
- **col**: The color of the points. Default to "black"
- **...**: additional unused argument

Author(s)
Nicoletta D’Angelo

See Also
stp, print.stp, summary.stp

Examples

code:  
```r
set.seed(2)
df <- data.frame(cbind(runif(100), runif(100), runif(100)))

stp1 <- stp(df)
#plot
plot(stp1)

#cumulative time occurrences
plot(stp1, tcum = FALSE)

#change color of points
plot(stp1, col = "blue")

#display only in space-time
```

plot(stp1, marg = FALSE)

#discrete times
set.seed(2)
stp2 <- stp(data.frame(cbind(runif(100), runif(100), round(runif(100) * 100))))
plot(stp2)

plot.stpm
Plot a stpm object

Description
This function plots the marked point pattern stored in the stpm object given in input, in a three panel plot representing the 3Dplot of the coordinates, and the mark values.

Usage
## S3 method for class 'stpm'
plot(x, ...)

Arguments
x An object of class stpm
...
additional unused argument

Author(s)
Nicoletta D’Angelo

See Also
stppm

Examples

df <- data.frame(cbind(runif(100), runif(100), runif(100), rpois(100, 15),
rpois(100, 30)))

stp1 <- stpm(df)
plot(stp1)

## Categorical marks
dfA <- data.frame(x = runif(100), y = runif(100), t = runif(100),
ml = rnorm(100), m2 = rep("C", times = 100))
dfB <- data.frame(x = runif(50), y = runif(50), t = runif(50), 
m1 = rnorm(25), m2 = rep("D", times = 50))

stpm2 <- stpm(rbind(dfA, dfB), names = c("continuous", "dichotomous"))

plot(stpm2)

---

**plot.stppm**  
*Plot of the fitted intensity of a spatio-temporal Poisson process model*

**Description**

The function plots the fitted intensity, displayed both in space and in space and time.

**Usage**

```r
## S3 method for class 'stppm'
plot(
  x, 
  scaler = c("silverman", "IQR", "sd", "var"),
  do.points = TRUE,
  print.bw = FALSE,
  zap = 1e-05,
  par = TRUE,
  ...
)
```

**Arguments**

- **x**: An object of class stppm
- **scaler**: Optional. Controls the value for a scalar representation of the spatial scale of the data. Either a character string, "silverman" (default), "IQR", "sd", or "var"; or positive numeric value(s). See OS.
- **do.points**: Add points to plot
- **print.bw**: It prints the estimated oversmoothing (OS) bandwidth selector
- **zap**: Noise threshold factor (default to 0.00001). A numerical value greater than or equal to 1. If the range of pixel values is less than zap * .Machine$double.eps, the image will be treated as constant. This avoids displaying images which should be constant but contain small numerical errors.
- **par**: Default to TRUE.
- **...**: additional unused argument

**Author(s)**

Nicoletta D’Angelo and Giada Adelfio
References


See Also

stppm, print.stppm, summary.stppm

Examples

```r
set.seed(2)
pin <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(2, 6),
nsim = 1, verbose = TRUE)
inh1 <- stppm(pin, formula = ~ x)
plot(inh1)
```

---

**print.globaldiag**  
*Print global diagnostics of a spatio-temporal point process first-order intensity*

Description

This function performs global diagnostics of a model fitted for the first-order intensity of a spatio-temporal point pattern, by returning the sum of the squared differences between the estimated and the theoretical K-functions obtained through globaldiag.

Usage

```r
## S3 method for class 'globaldiag'
print(x, ...)
```

Arguments

- `x`: A globaldiag object
- `...`: additional unused argument
Value

It returns the sum of the squared differences between the estimated and the theoretical K-functions obtained through `globaldiag`.

Author(s)

Nicoletta D’Angelo

References


See Also

globaldiag, plot.globaldiag, summary.globaldiag

Examples

```R
set.seed(2)
inherited <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
                    par = c(3, 6))

mod1 <- stppm(inherited, formula = ~ 1)
mod2 <- stppm(inherited, formula = ~ x)

g1 <- globaldiag(inherited, mod1$s1)
g2 <- globaldiag(inherited, mod2$s1)

g1

g2
```
print.lista  
*Print a lista object*

**Description**

It prints the main information on the local network summary statistics stored in a lista object.

**Usage**

```r
## S3 method for class 'lista'
print(x, ...)
```

**Arguments**

- `x`  
  An object of class lista
- `...`  
  additional unused argument

**Author(s)**

Nicoletta D'Angelo

**Examples**

```r
set.seed(2)
df_net <- data.frame(x = runif(25, 0, 0.85), y = runif(25, 0, 0.85), t = runif(25))
stlp1 <- stp(df_net, L = chicagonet)
lambda <- rep(diff(range(stlp1$df$x)) * diff(range(stlp1$df$y)) * diff(range(stlp1$df$t)) / spatstat.geom::volume(stlp1$L),
nrow(stlp1$df))

k <- localSTLKinhom(stlp1, lambda = lambda, normalize = TRUE)

k
```

---

print.localdiag  
*Print of the diagnostics' result on a spatio-temporal point process model*

**Description**

It prints the main information on the result of the local diagnostics performed with localdiag on either a stp or stlp object: whether the local test was run on point patterns lying on a linear network or not; the number of points in the analysed spatio-temporal point pattern $X$; the number of points of $X$ which are identified as outlying by the previously performed local diagnostics.
print.localdiag

Usage

## S3 method for class 'localdiag'
print(x, ...)

Arguments

x       A localdiag object
...
    additional unused argument

Author(s)

Nicoletta D’Angelo and Giada Adelfio

References

Adelfio, G., Siino, M., Mateu, J., and Rodríguez-Cortés, F. J. (2020). Some properties of local
weighted second-order statistics for spatio-temporal point processes. Stochastic Environmental Re-
search and Risk Assessment, 34(1), 149-168.

D’Angelo, N., Adelfio, G. and Mateu, J. (2022) Local inhomogeneous second-order characteristics
for spatio-temporal point processes on linear networks. Stat Papers. https://doi.org/10.1007/s00362-
022-01338-4

See Also

infl, plot.localdiag, summary.localdiag

Examples

set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
          par = c(.3, 6))
mod1 <- stppm(inh, formula = ~ 1)
resmod1 <- localdiag(inh, mod1$l, p = .9)
resmod1
\section*{print.localtest \hfill \textit{Print of the result of the permutation local test}}

\subsection*{Description}

It prints the main information on the result of the local permutation test performed with \texttt{localtest} on either a \texttt{stp} or \texttt{stlp} object: whether the local test was run on point patterns lying on a linear network or not; the number of points in the background $X$ and alternative $Z$ patterns; the number of points in $X$ which exhibit local differences in the second-order structure with respect to $Z$, according to the performed test.

\subsection*{Usage}

\begin{verbatim}
## S3 method for class 'localtest'
print(x, ...)
\end{verbatim}

\subsection*{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{0.5cm} An object of class \texttt{localtest}
  \item \texttt{...} \hspace{0.5cm} additional unused argument
\end{itemize}

\subsection*{Author(s)}
Nicoletta D’Angelo

\subsection*{References}


\subsection*{See Also}
\texttt{localtest}, \texttt{summary.localtest}, \texttt{plot.localtest}

\subsection*{Examples}

\begin{verbatim}
set.seed(2)
X <- rstpp(lambda = function(x, y, t, a) (exp(a[1] + a[2]*x)),
          par = c(.005, 5))
Z <- rstpp(lambda = 30)

test <- localtest(X, Z, method = "K", k = 3)
\end{verbatim}
print.locstppm

Description

The function prints the main information of the distribution of the parameters of a fitted local spatio-temporal Poisson process model.

Usage

## S3 method for class 'locstppm'
print(x, ..., 

Arguments

x

An object of class locstppm

...

additional unused argument

Author(s)

Nicoletta D’Angelo

References


See Also

locstppm, summary.locstppm, plot.locstppm

Examples

set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
par = c(0.005, 5))
inh_local <- locstppm(inh, formula = ~ x)

inh_local
Print of a fitted separable spatio-temporal Poisson process model on a linear network

Description

The function prints the main information of the fitted model.

Usage

```r
## S3 method for class 'sepstlppm'
print(x, ...)
```

Arguments

- `x` An object of class `sepstlppm`
- `...` additional unused argument

Author(s)

Nicoletta D’Angelo

See Also

`sepstlppm`

Examples

```r
crimesub <- stpm(valenciacrimes$df[101:200, ],
    names = colnames(valenciacrimes$df)[-c(1:3)],
    L = valencianet)

mod1 <- sepstlppm(crimesub, spaceformula = ~x,
    timeformula = ~ day)

mod1
```
Print of a fitted separable spatio-temporal Poisson process model

Description

The function prints the main information of the fitted model.

Usage

```r
## S3 method for class 'sepstppm'
print(x, ...)
```

Arguments

- `x` An object of class `sepstppm`
- `...` additional unused argument

Author(s)

Nicoletta D’Angelo

See Also

`sepstppm`

Examples

```r
crimesub <- stpm(valenciacrimes$df[101:200, ],
    names = colnames(valenciacrimes$df)[-c(1:3)])

mod1 <- sepstppm(crimesub, spaceformula = ~x ,
    timeformula = ~ day)

mod1
```
print.stcov  
*Print a stcov object*

**Description**

It prints the main information on the spatio-temporal covariate stored in the stcov object: the number of points; the enclosing spatial window; the temporal time period; information on the covariate values.

**Usage**

```r
## S3 method for class 'stcov'
print(x, ...)
```

**Arguments**

- `x` An object of class stcov
- `...` additional unused argument

**Author(s)**

Nicoletta D’Angelo

**Examples**

```r
set.seed(2)
df <- data.frame(runif(100), runif(100), runif(100), rpois(100, 15))
cov <- stcov(df, interp = FALSE)
cov
```

print.stlgcppm  
*Print of a fitted LGCP model*

**Description**

The function prints the main information on the fitted model. In this case of local parameters (both first- and second-order), the summary function contains information on their distributions.

**Usage**

```r
## S3 method for class 'stlgcppm'
print(x, ...)
```
Arguments

x           An object of class stlgcppm
...         additional unused argument

Author(s)

Nicoletta D’Angelo and Giada Adelfio

References


See Also

stlgcppm, print.stlgcppm, localsummary, plot.stlgcppm, localplot

Examples

catsub <- stp(greececatalog$df[1:200, ])
lgcp1 <- stlgcppm(catsub)
lgcp1

print.stlp

Print a stlp object

Description

It prints the main information on the spatio-temporal point pattern on a linear network stored in the stlp object: the number of points; vertices and lines of the linear network; the enclosing spatial window; the temporal time period.

Usage

## S3 method for class 'stlp'
print(x, ...)

# Example usage:
catsub <- stp(greececatalog$df[1:200, ])
lgcp1 <- stlgcppm(catsub)
print(lgcp1)
Arguments

x An object of class stlp

... additional unused argument

Author(s)

Nicoletta D’Angelo

See Also

stp, plot.stlp, summary.stlp

Examples

set.seed(2)
df_net <- data.frame(cbind(runif(100, 0, 0.85), runif(100, 0, 0.85), runif(100)))

stlp1 <- stp(df_net, L = chicagonet)
stlp1

Description

It prints the main information on the spatio-temporal point pattern stored in the stlp object: the number of points; the enclosing spatial window; the temporal time period; information on marks.

Usage

## S3 method for class 'stlp'
print(x, ...)

Arguments

x An object of class stlp

... additional unused argument

Author(s)

Nicoletta D’Angelo
print.stp

Examples

```r
set.seed(2)
df <- data.frame(x = runif(100, 0, 0.8), y = runif(100, 0, 0.8), t = runif(100), m = rpois(100, 15))
stlpm1 <- stpm(df, L = chicagonet)
stlpm1
```

---

print.stp  

Print a stp object

Description

It prints the main information on the spatio-temporal point pattern stored in the stp object: the number of points; the enclosing spatial window; the temporal time period.

Usage

```r
## S3 method for class 'stp'
print(x, ...)
```

Arguments

- `x`  
  An object of class stp

- `...`  
  additional unused argument

Author(s)

Nicoletta D’Angelo

See Also

stp, summary.stp, plot.stp

Examples

```r
set.seed(2)
df <- data.frame(cbind(runif(100), runif(100), runif(100)))

stp1 <- stp(df)
stp1
```
### print.stpm

**Print a stpm object**

**Description**

It prints the main information on the spatio-temporal point pattern stored in the stpm object: the number of points; the enclosing spatial window; the temporal time period; information on marks.

**Usage**

```r
## S3 method for class 'stpm'
print(x, ...)
```

**Arguments**

- `x` An object of class stpm
- `...` additional unused argument

**Author(s)**

Nicoletta D’Angelo

**Examples**

```r
set.seed(2)
df <- data.frame(cbind(runif(100), runif(100), runif(100), rpois(100, 15),
rpois(100, 30)))
stpm1 <- stpm(df)
summary(stpm1)
```

---

### print.stppm

**Print of a fitted spatio-temporal Poisson process model**

**Description**

The function prints the main information of the fitted model.

**Usage**

```r
## S3 method for class 'stppm'
print(x, ...)
```
**Arguments**

- `x` An object of class `stppm`
- `...` additional unused argument

**Author(s)**

Nicoletta D’Angelo

**References**


**See Also**

`stppm, print.stppm, plot.stppm`

**Examples**

```r
cat <- set.seed(2)
pin <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(2, 6))
in1h <- stppm(pin, formula =~ x)
in1h
```

---

**Description**

This function simulates a spatio-temporal ETAS (Epidemic Type Aftershock Sequence) process on a linear network as a `stpm` object.

It is firstly introduced and employed for simulation studies in D’Angelo et al. (2021).

It follows the generating scheme for simulating a pattern from an Epidemic Type Aftershocks-Sequences (ETAS) process (Ogata and Katsura 1988) with conditional intensity function (CIF) as in Adelfio and Chiodi (2020), adapted for the space location of events to be constrained on a linear network.

The simulation on the network is guaranteed by the homogeneous spatial Poisson processes being generated on the network.
Usage

rETASlp(
  pars = NULL,
  betacov = 0.39,
  m0 = 2.5,
  b = 1.0789,
  tmin = 0,
  t.lag = 200,
  covsim = FALSE,
  L,
  all.marks = FALSE
)

Arguments

pars           A vector of parameters of the ETAS model to be simulated. See the 'Details'
               section.
betacov        Numerical array. Parameters of the covariates ETAS model
m0             Parameter for the background general intensity of the ETAS model. In the com-
               mon seismic analyses it represents the threshold magnitude.
b             1.0789
Tmin           Minimum value of time.
t.lag           200
Covsim         Default FALSE
L              linear network
All.marks      Logical value indicating whether to store all the simulation information as marks
                in the stlpm object. If FALSE (default option) only the magnitude is returned.

Details

The CIF of an ETAS process as in Adelfio and Chiodi (2020) can be written as

$$\lambda_\theta(t, u|H_t) = \mu f(u) + \sum_{i_j < t} \frac{\kappa_0 \exp(\eta_j)}{(t - t_j + c)^p (u - u_j)^2 + d}^{-q},$$

where

- $H_t$ is the past history of the process up to time $t$
- $\mu$ is the large-scale general intensity
- $f(u)$ is the spatial density
- $\eta_j = \beta' Z_j$ is a linear predictor
- $Z_j$ the external known covariate vector, including the magnitude
- $\theta = (\mu, \kappa_0, c, p, d, q, \beta)$ are the parameters to be estimated
- $\kappa_0$ is a normalising constant
- $c$ and $p$ are characteristic parameters of the seismic activity of the given region,
and $d$ and $q$ are two parameters related to the spatial influence of the mainshock

In the usual ETAS model for seismic analyses, the only external covariate represents the magnitude, $\beta = \alpha$, as $\eta_j = \beta' Z_j = \alpha(m_j - m_0)$, where $m_j$ is the magnitude of the $j^{th}$ event and $m_0$ the threshold magnitude, that is, the lower bound for which earthquakes with higher values of magnitude are surely recorded in the catalogue.

Value

A stpm object

Author(s)

Nicoletta D’Angelo and Marcello Chiodi

References


Examples

```r
set.seed(95)
X <- rETASlp(pars = c(0.1293688525, 0.003696, 0.013362, 1.2, 0.424466, 1.164793),
             L = chicagonet)
```

Description

This function simulates a spatio-temporal ETAS (Epidemic Type Aftershock Sequence) process as a stpm object.

It follows the generating scheme for simulating a pattern from an Epidemic Type Aftershocks-Sequences (ETAS) process (Ogata and Katsura 1988) with conditional intensity function (CIF) as in Adelfio and Chiodi (2020), adapted for the space location of events to be constrained.

See the 'Details' section.
Usage

\texttt{rETASp(pars = NULL, betacov = 0.39, m0 = 2.5, b = 1.0789, tmin = 0, t.lag = 200, xmin = 0, xmax = 1, ymin = 0, ymax = 1, covsim = FALSE, all.marks = FALSE)}

Arguments

\begin{itemize}
  \item \texttt{pars} A vector of parameters of the ETAS model to be simulated. See the 'Details' section.
  \item \texttt{betacov} Numerical array. Parameters of the ETAS model covariates.
  \item \texttt{m0} Parameter for the background general intensity of the ETAS model. In the common seismic analyses it represents the threshold magnitude.
  \item \texttt{b} 1.0789
  \item \texttt{tmin} Minimum value of time.
  \item \texttt{t.lag} 200
  \item \texttt{xmin} Minimum of x coordinate range
  \item \texttt{xmax} Maximum of x coordinate range
  \item \texttt{ymin} Minimum of y coordinate range
  \item \texttt{ymax} Maximum of y coordinate range
  \item \texttt{covsim} Default FALSE
  \item \texttt{all.marks} Logical value indicating whether to store all the simulation information as marks in the \texttt{stpm} object. If FALSE (default option) only the magnitude is returned.
\end{itemize}

Details

The CIF of an ETAS process as in Adelfio and Chiodi (2020) can be written as

\[ \lambda_\theta(t, u|\mathcal{H}_t) = \mu f(u) + \sum_{t_j < t} \frac{\kappa_0 \exp(\eta_j)}{(t - t_j + e)^p} \{(u - u_j)^2 + d\}^{-q}, \]

where

- \( \mathcal{H}_t \) is the past history of the process up to time \( t \)
- \( \mu \) is the large-scale general intensity
$f(u)$ is the spatial density

$\eta_j = \beta' Z_j$ is a linear predictor

$Z_j$ the external known covariate vector, including the magnitude

$\theta = (\mu, \kappa_0, c, p, d, q, \beta)$ are the parameters to be estimated

$\kappa_0$ is a normalising constant

c and p are characteristic parameters of the seismic activity of the given region, and d and q are two parameters related to the spatial influence of the mainshock

In the usual ETAS model for seismic analyses, the only external covariate represents the magnitude, $\beta = \alpha$, as $\eta_j = \beta' Z_j = \alpha(m_j - m_0)$, where $m_j$ is the magnitude of the $j^{th}$ event and $m_0$ the threshold magnitude, that is, the lower bound for which earthquakes with higher values of magnitude are surely recorded in the catalogue.

Value

A stpm object

Author(s)

Nicoletta D’Angelo and Marcello Chiodi

References


Examples

```r
set.seed(95)
X <- rETASp(pars = c(0.1293688525, 0.003696, 0.013362, 1.2,0.424466, 1.164793),
            betacov = 0.5,
            xmin = 600, xmax = 2200, ymin = 4000, ymax = 5300)
plot(X)
```

rstlpp

| Simulate homogeneous and inhomogeneous spatio-temporal Poisson point patterns on linear networks |

Description

This function creates a stlp object, simulating a spatio-temporal point pattern on a linear network following either an homogeneous or inhomogeneous intensity.
Usage

```r
rstlpp(
  lambda = 500,
  nsim = 1,
  verbose = FALSE,
  par = NULL,
  minX = 0,
  maxX = 1,
  minY = 0,
  maxY = 1,
  minT = 0,
  maxT = 1,
  L
)
```

Arguments

- `lambda`: Expected number of points to simulate
- `nsim`: Number of patterns to simulate. Default to 1.
- `verbose`: Default to `FALSE`
- `par`: Parameters of the reference intensity
- `minX`: Minimum of x coordinate range
- `maxX`: Maximum of x coordinate range
- `minY`: Minimum of y coordinate range
- `maxY`: Maximum of y coordinate range
- `minT`: Minimum of t coordinate range
- `maxT`: Maximum of t coordinate range
- `L`: linear network

Value

A `stp` object

Author(s)

Nicoletta D’Angelo

Examples

```r
set.seed(2)
h1 <- rstlpp(lambda = 500, L = chicagonet)

set.seed(2)
h1 <- rstlpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(4, 1.5), L = chicagonet)
```
rstpp

Simulate homogeneous and inhomogeneous spatio-temporal Poisson point patterns

Description

This function creates a stp object, simulating a spatio-temporal point pattern following either an homogeneous or inhomogeneous intensity

Usage

rstpp(
  lambda = 500,
  nsim = 1,
  verbose = FALSE,
  par = NULL,
  minX = 0,
  maxX = 1,
  minY = 0,
  maxY = 1,
  minT = 0,
  maxT = 1
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lambda</td>
<td>Expected number of points to simulate</td>
</tr>
<tr>
<td>nsim</td>
<td>Number of patterns to simulate. Default to 1.</td>
</tr>
<tr>
<td>verbose</td>
<td>Default to FALSE</td>
</tr>
<tr>
<td>par</td>
<td>Parameters of the reference intensity</td>
</tr>
<tr>
<td>minX</td>
<td>Minimum of x coordinate range</td>
</tr>
<tr>
<td>maxX</td>
<td>Maximum of x coordinate range</td>
</tr>
<tr>
<td>minY</td>
<td>Minimum of y coordinate range</td>
</tr>
<tr>
<td>maxY</td>
<td>Maximum of y coordinate range</td>
</tr>
<tr>
<td>minT</td>
<td>Minimum of t coordinate range</td>
</tr>
<tr>
<td>maxT</td>
<td>Maximum of t coordinate range</td>
</tr>
</tbody>
</table>

Value

A stp object

Author(s)

Nicoletta D’Angelo
See Also

stppm

Examples

# homogeneous Poisson processes
set.seed(2)
h1 <- rstpp(lambda = 500)

set.seed(2)
h2 <- rstpp(lambda = 500, minX = 0,
    maxX = 2, minY = 3, maxY = 5, minT = 1, maxT = 9)

set.seed(2)
h3 <- rstpp(lambda = 900, nsim = 3, verbose = TRUE)

# inhomogeneous Poisson process
set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(2, 6))

sepstlppm

Fit a separable spatio-temporal Poisson process model on a linear network

Description

Fit a separable spatio-temporal Poisson process model on a linear network

Usage

sepstlppm(x, spaceformula, timeformula)

Arguments

x          A stlpm object
spaceformula A formula for the spatial component. See lppm for details
timeformula A formula for the temporal component. It fits a log-linear model with the glm function

Value

An object of class sepstlppm
sepstppm

Examples

\[
\text{crimesub} \leftarrow \text{stpm(valenciacrimes}$\text{df}[101:200, ],
\text{names} = \text{colnames(valenciacrimes}$\text{df}[-c(1:3)],
\text{L} = \text{valencianet})
\]

\[
\text{mod1} \leftarrow \text{sepstlppm(crimesub, spaceformula = } ~x ,
\text{timeformula = } ~\text{day})
\]

sepstppm  
Fit a separable spatio-temporal Poisson process model

Description
Fit a separable spatio-temporal Poisson process model

Usage

\[
\text{sepstppm}(x, \text{spaceformula, timeformula})
\]

Arguments

\begin{itemize}
\item \text{x} \quad \text{A \text{stpm} object}
\item \text{spaceformula} \quad \text{A formula for the spatial component. See \text{ppm} for details}
\item \text{timeformula} \quad \text{A formula for the temporal component. It fits a log-linear model with the \text{glm} function}
\end{itemize}

Value

An object of class sepstppm

Examples

\[
\text{crimesub} \leftarrow \text{stpm(valenciacrimes}$\text{df}[101:200, ],
\text{names} = \text{colnames(valenciacrimes}$\text{df}[-c(1:3)])
\]

\[
\text{mod1} \leftarrow \text{sepstppm(crimesub, spaceformula = } ~x ,
\text{timeformula = } ~\text{day})
\]
stcov

Create stcov objects and interpolate spatio-temporal covariates on a regular grid

Description

This function interpolates the covariate values observed at some observed sites to a regular grid. The input object should be either a matrix or a dataframe with four columns: x, y, t, and the covariate values, named as the covariate later called in the model formula (see stppm). The interpolation is performed through Inverse Distance Weighting (IDW). See the Details.

Usage

stcov(
  x,
  interp = TRUE,
  nx = NULL,
  mult = 1,
  p = 81,
  names = NULL,
  verbose = FALSE
)

Arguments

x
A data.frame with four columns, containing the spatio-temporal coordinates and the covariate values.

interp
Logical value indicating whether to interpolate the covariate on a regular grid. Default to TRUE.

nx
Number of coordinates to generate for each dimension. The default is floor((mult * nrow(cov)) ^ (1/3)).

mult
The multiplicand of the number of points in the default for nx.

p
Power of IDW distances.
	names
Factor string to name the covariate.

verbose
Default to FALSE. If TRUE, the elapsed minutes are printed.

Details

The function builds a regular grid with equispaced values along the three coordinates and interpolates the covariate values at the new locations. The interpolation at a point location \( x_k \) is performed through the inverse-distance weighting smoothing procedure of the covariate values \( Z(x_j) \) at their sampling locations \( j = 1, \ldots, J \). In such a case, the smoothed value at location \( x_k \) is

\[
Z(x_k) = \frac{\sum_j w_j Z(x_j)}{\sum_j w_j},
\]
where the weight \( w_j \) is the \( j \)-th element of the inverse \( p \)th powers of distance,

\[
w = \{ w_j \}_{j=1}^{J} = \{ \frac{1}{d(x_k - x_j)^p} \}_{j=1}^{J},
\]

with

\[
d(x_k - x_j) = ||x_k - x_j||
\]

the Euclidean distance from \( x_k \) to \( x_j \).

Value

A \texttt{stpm} object, to be imputed as list object in \texttt{stppm}.

Author(s)

Nicoletta D’Angelo

See Also

\texttt{stppm}

Examples

```r
set.seed(2)
df <- data.frame(runif(100), runif(100), runif(100), rpois(100, 15))

cov <- stcov(df, interp = FALSE)
```

---

**stlgcppm**

Fit a log-Gaussian Cox process model to a spatio-temporal point pattern

Description

This function estimates a log-Gaussian Cox process (LGCP), following the **joint minimum contrast** procedure introduced in Siino et al. (2018).

Three covariances are available: separable exponential, Gneiting, and De Iaco-Cesare.

If the first and second arguments are set to \texttt{local}, a local log-Gaussian Cox process is fitted by means of the **locally weighted minimum contrast** procedure proposed in D’Angelo et al. (2023).
Usage

stlgcppm(
  X,
  formula = ~1,
  verbose = TRUE,
  seed = NULL,
  cov = c("separable", "gneiting", "iaco-cesare"),
  first = c("global", "local"),
  second = c("global", "local"),
  mult = 4,
  hs = c("global", "local"),
  npx0 = 10,
  npt0 = 10,
  itnmax = 100,
  min_vals = NULL,
  max_vals = NULL
)

Arguments

X A stp object

formula An object of class formula: a symbolic description of the first-order intensity to be fitted. The current version only supports formulas depending on the spatial and temporal coordinates: x, y, t. Default to formula = ~ 1 which provides an homogeneous first-order intensity.

verbose Default to TRUE

seed The seed used for the simulation of the dummy points. Default to NULL.

cov Covariance function to be fitted for the second-order intensity function. Default to separable. Other options are gneiting and iaco-cesare".

first Character string indicating whether to fit a first-order intensity function with global or local parameters: either global (default) or local.

second Character string indicating whether to fit a second-order intensity function with global or local parameters: either global (default) or local.

mult The multiplicand of the number of data points, for setting the number of dummy points to generate for the quadrature scheme

hs Character string indicating whether to select fixed or variable bandwidths for the kernel weights to be used in the log-likelihood. In any of those cases, the well-supported rule-of-thumb for choosing the bandwidth of a Gaussian kernel density estimator is employed. If hs = "global" (default), a fixed bandwidth is selected. If hs = "local", an individual bandwidth is selected for each point in the pattern X.

npx0 A positive integer representing the spatial distance to np-th closest event. Used in the computation of the local bandwidth. Suitable values are in the range from 10 (default) to 100.
npt0  A positive integer representing the temporal distance to np-th closest event. Used in the computation of the local bandwidth. Suitable values are in the range from 10 (default) to 100.

itnmax  Maximum number of iterations to run in the optimization procedure for the estimation of the second-order intensity parameters.

min_vals  Minimum values of the optimization procedure for the minimum contrast.

max_vals  Maximum values of the optimization procedure for the minimum contrast.

Details
Following the inhomogeneous specification in Diggle et al. (2013), we consider LGCPs with intensity
\[
\Lambda(u, t) = \lambda(u, t) \exp(S(u, t)).
\]

Value
A list of the class `stlgcppm`, containing

- `IntCoefs`  The fitted coefficients of the first-order intensity function
- `CovCoefs`  The fitted coefficients of the second-order intensity function
- `X`  The stp object provided as input
- `formula`  The formula provided as input
- `cov`  A string with the chosen covariance type
- `l`  Fitted first-order intensity
- `mu`  Mean function of the random intensity
- `mod_global`  The glm object of the model fitted to the quadrature scheme for the first-order intensity parameters estimation
- `newdata`  The data used to fit the model, without the dummy points
- `time`  Time elapsed to fit the model, in minutes

Author(s)
Nicoletta D’Angelo, Giada Adelfio, and Marianna Siino

References


**See Also**

`print.stlgcppm, summary.stlgcppm, localsummary, plot.stlgcppm, localplot`

**Examples**

```r
catsub <- stp(greececatalog$df[1:200, ])
lgcpl <- stlgcppm(catsub)
```

---

### `stp`

*Create stp and stlp objects for point patterns storage*

**Description**

This function creates a `stp` object as a dataframe with three columns: `x`, `y`, and `t`. If also the linear network `L`, of class `linnet`, is provided, a `stlp` object is created instead.

**Usage**

```r
stp(df, L)
```

**Arguments**

- `df`: A matrix with three columns, containing two space and the temporal coordinates
- `L`: Optional. The linear network of class `linnet`. If provided, the function returns a `stlp` object.

**Value**

An `stp` or `stlp` object, depending on whether or not an object of class `linnet` is provided for the `L` argument.

**Author(s)**

Nicoletta D’Angelo
**stpm**

See Also

- `summary.stp`, `print.stp`, `plot.stp`
- `stppm`, `print.stp`, `summary.stp`, `plot.stp`, `print.stlp`, `summary.stlp`, `plot.stlp`

Examples

```r
set.seed(2)
df <- data.frame(runif(100), runif(100), runif(100))
stp1 <- stp(df)
set.seed(2)
df_net <- data.frame(runif(100, 0, 0.85), runif(100, 0, 0.85), runif(100))
stlp1 <- stp(df_net, L = chicagonet)
```

---

**stpm**

Create stpm and stlpm objects for marked point patterns storage

### Description

This function creates a stpm object as a dataframe with 3 + m columns: x, y, t, and m columns to store different marks. If also the linear network L, of class linnet, is provided, a stlp object is created instead.

### Usage

```r
stpm(df, names = NULL, L)
```

### Arguments

- `df` A matrix with three columns + m marks
- `names` Factor string to name the marks columns.
- `L` Optional. The linear network of class linnet. If provided, the function returns a stlp object.

### Value

An stpm or stlpm object, depending on whether or not an object of class linnet is provided for the L argument.

### Author(s)

Nicoletta D’Angelo
Examples

```r
set.seed(2)
df <- data.frame(cbind(runif(100), runif(100), runif(100), rpois(100, 15),
rpois(100, 30)))

stpm1 <- stpm(df)

## Categorical marks
set.seed(2)
dfA <- data.frame(x = runif(100), y = runif(100), t = runif(100),
                  m1 = rnorm(100), m2 = rep(c("C"), times = 100))
dfB <- data.frame(x = runif(50), y = runif(50), t = runif(50),
                  m1 = rnorm(25), m2 = rep(c("D"), times = 50))

stpm2 <- stpm(rbind(dfA, dfB), names = c("continuous", "dichotomous"))

## Linear network
set.seed(2)
dfl <- data.frame(cbind(runif(100, 0, 0.85), runif(100, 0, 0.85), runif(100),
                       rpois(100, 15)))

stlpml <- stpm(dfl, L = chicagonet)
```

---

**stppm**

*Fit a Poisson process model to a spatio-temporal point pattern*

### Description

This function fits a Poisson process model to an observed spatio-temporal point pattern stored in a stp object.

### Usage

```r
stppm(
  X,  # spatio-temporal point pattern
  formula,  # formula for the intensity function
  formula_mark = NULL,  # formula for the mark intensity function
  covs = NULL,  # covariates for the intensity function
  marked = FALSE,  # whether the marks are binary
  spatial.cov = FALSE,  # whether the spatial covariates are binary
  verbose = FALSE,  # whether to print progress
  mult = 4,  # number of threads to use
  interp = TRUE,  # whether to interpolate the data
```


```r
parallel = FALSE,
sites = 1,
seed = NULL,
ncube = NULL,
grid = FALSE,
ncores = 2,
lsr = FALSE
)
```

**Arguments**

- **X**
  - A stp object

- **formula**
  - An object of class "formula": a symbolic description of the model to be fitted. The current version only supports formulas depending on the spatial and temporal coordinates: x, y, t.

- **formula.mark**
  - An object of class "formula"

- **covs**
  - A list containing stcov objects of possible spatio-temporal covariates. It is advisable to construct the stcov objects with stcov. Each stcov object should contain the spatio-temporal coordinates and the covariate values as the fourth column, named as the covariate called in the formula.

- **marked**
  - Logical value indicating whether the point process model to be fit is multitype. Default to FALSE.

- **spatial.cov**
  - Logical value indicating whether the point process model to be fit depends on spatio-temporal covariates. Default to FALSE.

- **verbose**
  - Default to FALSE.

- **mult**
  - The multiplicand of the number of data points, for setting the number of dummy points to generate for the quadrature scheme.

- **interp**
  - Logical value indicating whether to interpolate covariate values to dummy points or to use the covariates locations as dummies. Default to TRUE.

- **parallel**
  - Logical values indicating whether to use parallelization to interpolate covariates. Default to FALSE.

- **sites**
  - ..... 

- **seed**
  - The seed used for the simulation of the dummy points. Default to NULL.

- **ncube**
  - Number of cubes used for the cubature scheme.

- **grid**
  - Logical value indicating whether to generate dummy points on a regular grid or randomly. Default to FALSE.

- **ncores**
  - Number of cores to use, if parallelizing. Default to 2.

- **lsr**
  - Logical value indicating whether to use Logistic Spatio-Temporal Regression or Poisson regression. Default to FALSE.

**Details**

We assume that the template model is a Poisson process, with a parametric intensity or rate function \( \lambda(u, t; \theta) \) with space and time locations \( u \in W, t \in T \) and parameters \( \theta \in \Theta \).

Estimation is performed through the fitting of a glm using a spatio-temporal version of the quadrature scheme by Berman and Turner (1992).
Value
An object of class stppm. A list of

- `IntCoefs` The fitted coefficients
- `X` The `stp` object provided as input
- `nX` The number of points in `X`
- `I` Vector indicating which points are dummy or data
- `y_resp` The response variable of the model fitted to the quadrature scheme
- `formula` The formula provided as input
- `l` Fitted intensity
- `mod_global` The `glm` object of the model fitted to the quadrature scheme
- `newdata` The data used to fit the model, without the dummy points
- `time` Time elapsed to fit the model, in minutes

Author(s)
Nicoletta D’Angelo and Marco Tarantino

References

See Also
plot.stppm, print.stppm, summary.stppm
locstppm

Examples
```r
set.seed(2)
ph <- rstpp(lambda = 200)
hom1 <- stppm(ph, formula = ~ 1)

## Inhomogeneous
set.seed(2)
pin <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(2, 6))
inhl1 <- stppm(pin, formula = ~ x)

## Inhomogeneous depending on external covariates
```


```r
set.seed(2)
df1 <- data.frame(runif(100), runif(100), runif(100), rpois(100, 15))
df2 <- data.frame(runif(100), runif(100), runif(100), rpois(100, 15))

obj1 <- stcov(df1, names = "cov1")
obj2 <- stcov(df2, names = "cov2")

covariates <- list(cov1 = obj1, cov2 = obj2)

inh2 <- stppm(pin, formula = ~ x + cov2, covs = covariates, spatial.cov = TRUE)

## Inhomogeneous semiparametric

inh3 <- stppm(pin, formula = ~ s(x, k = 30))

## Multitype

set.seed(2)
dfA <- data.frame(x = runif(100), y = runif(100), t = runif(100),
m1 = rep(c("A"), times = 100))
dfB <- data.frame(x = runif(50), y = runif(50), t = runif(50),
m1 = rep(c("B"), each = 50))

stpm1 <- stpm(rbind(dfA, dfB))

inh4 <- stppm(stpm1, formula = ~ x + s(m1, bs = "re"), marked = TRUE)
```

---

**summary.globaldiag**  
*Summarizes global diagnostics of a spatio-temporal point process first-order intensity*

**Description**

This function performs global diagnostics of a model fitted for the first-order intensity of a spatio-temporal point pattern, by returning the sum of the squared differences between the estimated and the theoretical K-functions obtained through `globaldiag`.

**Usage**

```r
## S3 method for class 'globaldiag'
summary(object, ...)
```

**Arguments**

- `object` A `globaldiag` object
- `...` additional unused argument
Value

It returns the sum of the squared differences between the estimated and the theoretical K-functions obtained through `globaldiag`.

Author(s)

Nicoletta D’Angelo

References


See Also

`globaldiag`, `plot.globaldiag`, `summary.globaldiag`

Examples

```r
set.seed(2)
inr <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
            par = c(.3, 6))

mod1 <- stppm(inr, formula = ~ 1)
mod2 <- stppm(inr, formula = ~ x)

g1 <- globaldiag(inr, mod1$l)
g2 <- globaldiag(inr, mod2$l)

summary(g1)
summary(g2)
```
**summary.lista**  
*Summary a lista object*

**Description**

It prints the main information on the local network summary statistics stored in a `lista` object.

**Usage**

```r
## S3 method for class 'lista'
summary(object, ...)
```

**Arguments**

- `object`  
  An object of class `lista`

- `...`  
  additional unused argument

**Author(s)**

Nicoletta D'Angelo

**Examples**

```r
set.seed(2)
df_net <- data.frame(x = runif(25, 0, 0.85), y = runif(25, 0, 0.85), t = runif(25))
stlp1 <- stp(df_net, L = chicagonet)
lambda <- rep(diff(range(stlp1$df$x)) * diff(range(stlp1$df$y)) * diff(range(stlp1$df$t)) / spatstat.geom::volume(stlp1$L), nrow(stlp1$df))

k <- localSTLKinhom(stlp1, lambda = lambda, normalize = TRUE)

summary(k)
```

**summary.localdiag**  
*Summary of the diagnostics performed on a spatio-temporal point process model*

**Description**

It summarises the main information on the result of the local diagnostics performed with `localdiag` on either a `stp` or `stlp` object: whether the local test was run on point patterns lying on a linear network or not; the number of points in the analysed spatio-temporal point pattern X; the number of points of X which are identified as outlying by the previously performed local diagnostics.
Usage

```r
# S3 method for class 'localdiag'
summary(object, ...)  
```

Arguments

- `object`: A `localdiag` object
- `...`: additional unused argument

Author(s)

Nicoletta D’Angelo and Giada Adelfio

References


See Also

`infl`, `plot.localdiag`, `print.localdiag`

Examples

```r
set.seed(2)
inh <- rstep(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
             par = c(.3, 6))

mod1 <- stpp(inh, formula = ~ 1)
resmod1 <- localdiag(inh, mod1$l, p = .9)
summary(resmod1)
```
Description

It summarises the main information on the result of the local permutation test performed with `localtest` on either a `stp` or `stlp` object: whether the local test was run on point patterns lying on a linear network or not; the number of points in the background X and alternative Z patterns; the number of points in X which exhibit local differences in the second-order structure with respect to Z, according to the performed test.

Usage

```r
## S3 method for class 'localtest'
summary(object, ...)
```

Arguments

- `object`: An object of class `localtest`
- `...`: additional unused argument

Author(s)

Nicoletta D’Angelo

References


See Also

- `localtest`, `print.localtest`, `plot.localtest`

Examples

```r
set.seed(2)
X <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
           par = c(.005, 5))
Z <- rstpp(lambda = 30)

test <- localtest(X, Z, method = "K", k = 3)
```
Summary of a fitted local spatio-temporal Poisson process model

Description
The function summarises the main information on the distribution of the parameters of a fitted local spatio-temporal Poisson process model.

Usage
## S3 method for class 'locstppm'
summary(object, ...)

Arguments
object
An object of class locstppm
...
additional unused argument

Author(s)
Nicoletta D’Angelo

References

See Also
locstppm, print.locstppm, plot.locstppm

Examples
set.seed(2)
inh <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)},
par = c(0.005, 5))
inh_local <- locstppm(inh, formula = ~ x)
summary(inh_local)
**Description**

The function summarises the main information of the fitted model.

**Usage**

```r
## S3 method for class 'sepstlppm'
summary(object, ...)
```

**Arguments**

- `object` An object of class `sepstlppm`
- `...` additional unused argument

**Author(s)**

Nicoletta D'Angelo

**See Also**

`sepstlppm`

**Examples**

```r
crimesub <- stpm(valenciacrimes$df[101:200,],
               names = colnames(valenciacrimes$df)[-c(1:3)],
               L = valencianet)

mod1 <- sepstlppm(crimesub, spaceformula = ~x,
               timeformula = ~day)

summary(mod1)
```
summary.sepstppm

Summary of a fitted separable spatio-temporal Poisson process model

Description

The function summarises the main information of the fitted model.

Usage

## S3 method for class 'sepstppm'
summary(object, ...)

Arguments

object An object of class sepstppm

... additional unused argument

Author(s)

Nicoletta D’Angelo

See Also

sepstlppm

Examples

crimesub <- stpm(valenciacrimes$df[101:200, ],
                 names = colnames(valenciacrimes$df)[-c(1:3)])

mod1 <- sepstppm(crimesub, spaceformula = ~ x ,
                 timeformula = ~ day)

summary(mod1)
**summary.stcov**  
*Summary of a stcov object*

**Description**

It prints the summary statistics of the spatio-temporal coordinates and the covariates values of the spatio-temporal covariate stored in the stcov object.

**Usage**

```r
## S3 method for class 'stcov'
summary(object, ...)
```

**Arguments**

- `object` An object of class stcov
- `...` additional unused argument

**Author(s)**

Nicoletta D’Angelo

**Examples**

```r
set.seed(2)
df <- data.frame(runif(100), runif(100), runif(100), rpois(100, 15))
cov <- stcov(df, interp = FALSE)
summary(cov)
```

---

**summary.stlgcppm**  
*Summary of a fitted LGCP model*

**Description**

The function Summarises the main information on the fitted model. In this case of local parameters (both first- and second-order), the summary function contains information on their distributions.

**Usage**

```r
## S3 method for class 'stlgcppm'
summary(object, ...)
```
Arguments

object An object of class stlgcppm

Author(s)

Nicoletta D’Angelo and Giada Adelfio

References


See Also

stlgcppm, print.stlgcppm, localsummary, plot.stlgcppm, localplot

Examples

catsub <- stp(greecescatalog$df[1:200, ])
lgcpl <- stlgcppm(catsub)
summary(lgcpl)

summary.stlp

Summary of a stlp object

Description

It prints the main information on the spatio-temporal point pattern on a linear network stored in the stlp object: the number of points; vertices and lines of the linear network; the enclosing spatial window; the temporal time period.

Usage

## S3 method for class 'stlp'
summary(object, ...)
**Arguments**

- **object**  
  An object of class `stlp`
- ...  
  additional unused argument

**Author(s)**

Nicoletta D’Angelo

**See Also**

`stp`, `plot.stlp`, `print.stlp`

**Examples**

```r
set.seed(2)
df_net <- data.frame(cbind(runif(100, 0, 0.85), runif(100, 0, 0.85), runif(100)))

stlp1 <- stp(df_net, L = chicagonet)
summary(stlp1)
```

---

**summary.stlpm**  
*Summary of a stlpm object*

**Description**

It prints the summary statistics of the spatio-temporal coordinates and the marks of the spatio-temporal point pattern stored in the `stlpm` object.

**Usage**

```r
## S3 method for class 'stlpm'
summary(object, ...)
```

**Arguments**

- **object**  
  An object of class `stlpm`
- ...  
  additional unused argument

**Author(s)**

Nicoletta D’Angelo
Examples

```
set.seed(2)
df <- data.frame(x = runif(100, 0, 0.8), y = runif(100, 0, 0.8),
                 t = runif(100), m = rpois(100, 15))

stlpm1 <- stpm(df, L = chicagonet)

summary(stlpm1)
```

---

**summary.stp**  
*Summary of a stp object*

**Description**

It prints the summary statistics of the spatial and temporal coordinates of the spatio-temporal point pattern stored in the `stp` object.

**Usage**

```
## S3 method for class 'stp'
summary(object, ...)
```

**Arguments**

- `object`  
  An object of class `stp`

- `...`  
  additional unused argument

**Author(s)**

Nicoletta D’Angelo

**See Also**

`stp`, `print.stp`, `plot.stp`

**Examples**

```
set.seed(2)
df <- data.frame(cbind(runif(100), runif(100), runif(100)))

stp1 <- stp(df)

summary(stp1)
```
Description

It prints the summary statistics of the spatio-temporal coordinates and the marks of the spatio-temporal point pattern stored in the \texttt{stpm} object.

Usage

\begin{verbatim}
## S3 method for class 'stpm'
summary(object, ...)
\end{verbatim}

Arguments

- \texttt{object}: An object of class \texttt{stpm}
- \texttt{...}: additional unused argument

Author(s)

Nicoletta D’Angelo

Examples

\begin{verbatim}
set.seed(2)
df <- data.frame(cbind(runif(100), runif(100), runif(100), rpois(100, 15),
                      rpois(100, 30)))
stpm1 <- stpm(df)
summary(stpm1)

## Categorical marks
set.seed(2)
dfA <- data.frame(x = runif(100), y = runif(100), t = runif(100),
                  m1 = rnorm(100), m2 = rep(c("C"), times = 100))
dfB <- data.frame(x = runif(50), y = runif(50), t = runif(50),
                  m1 = rnorm(25), m2 = rep(c("D"), times = 50))
stpm2 <- stpm(rbind(dfA, dfB), names = c("continuous", "dichotomous"))
summary(stpm2)
\end{verbatim}
Summary.stppm

Summary of a fitted spatio-temporal Poisson process model

Description

The function summarises the main information of the fitted model.

Usage

```r
## S3 method for class 'stppm'
summary(object, ...)  
```

Arguments

- `object`: An object of class `stppm`
- `...`: additional unused argument

Author(s)

Nicoletta D’Angelo

References


See Also

`stppm`, `print.stppm`, `plot.stppm`

Examples

```r
set.seed(2)
pin <- rstpp(lambda = function(x, y, t, a) {exp(a[1] + a[2]*x)}, par = c(2, 6))
inh1 <- stppm(pin, formula = ~ x)

summary(inh1)
```
valenciacrimes

Crimes in Valencia in 2019

Description
A dataset in stpm format containing the 10929 crimes occurred in Valencia, Spain, in 2019.

Usage
data(valenciacrimes)

Format
A stpm object

Details
The 15 available marks are the following:

- month.
- week.
- day.
- week_day.
- atm_dist.
- bank_dist.
- bar_dist.
- cafe_dist.
- industrial_dist.
- market_dist.
- nightclub_dist.
- police_dist.
- pub_dist.
- restaurant_dist.
- taxi_dist.

Author(s)
Nicoletta D’Angelo

Examples
data(valenciacrimes)
valencianet

Roads of Valencia, Spain

Description
A linear network of class linnet of the roads of Valencia, Spain

Usage
data(valencianet)

Format
A linear network of class linnet

Author(s)
Nicoletta D’Angelo

Examples
data(valencianet)
Index

* datasets
  chicagonet, 4
  greececatalog, 6
  valenciacrimes, 81
  valencianet, 82

  chicagonet, 4

  glm, 56, 57
  globaldiag, 5, 10, 21, 37, 68
  greececatalog, 6

  infl, 8, 9, 10, 24, 39, 70

  KLISTAhat, 9, 17
  LISTAhat, 17
  localdiag, 8, 9, 23, 38, 69
  localplot, 11, 30, 45, 62, 76
  localSTLginhom, 12, 13, 14, 17, 22
  localSTLKinhom, 9, 13, 17, 22
  localsummary, 15, 30, 45, 62, 76
  localtest, 16, 24, 25, 40, 71
  locstppm, 11, 16, 18, 26, 41, 66, 72
  lppm, 56

  OS, 15, 26, 28, 30, 35

  plot, 9
  plot.globaldiag, 6, 20, 37, 68
  plot.lista, 22
  plot.localdiag, 8, 10, 23, 39, 70
  plot.localtest, 18, 24, 40, 71
  plot.locstppm, 25, 41, 72
  plot.sepstlppm, 27
  plot.sepstppm, 27
  plot.stcov, 29
  plot.stlgcppm, 29, 45, 62, 76
  plot.stlp, 31, 46, 63, 77
  plot.stlpm, 32
  plot.stp, 33, 47, 63, 78
  plot.stpm, 34
  plot.stppm, 35, 49, 66, 80
  plot3D, 31
  ppm, 57
  print, 9
  print.globaldiag, 6, 21, 36
  print.lista, 38
  print.localdiag, 8, 10, 24, 38, 70
  print.localtest, 18, 25, 40, 71
  print.locstppm, 26, 41, 72
  print.sepstlppm, 42
  print.sepstppm, 43
  print.stcov, 44
  print.stlgcppm, 30, 44, 45, 62, 76
  print.stlpm, 46
  print.stp, 33, 47, 63, 78
  print.stpm, 48
  print.stppm, 36, 48, 49, 66, 80
  rETASlp, 49
  rETASp, 51
  rstlpp, 53
  rstpp, 55

  sepstlppm, 42, 56, 73, 74
  sepstppm, 43, 57
  stcov, 29, 58
  STIKhat, 5
  stlgcppm, 11, 16, 30, 45, 59, 76
  STLginhom, 13, 14
  STLKinhom, 5, 13, 14, 21
  stopp (stopp-package), 3
  stopp-package, 3
  stp, 31, 33, 46, 47, 62, 77, 78
  stpm, 63
  stppm, 20, 32, 34, 36, 49, 56, 59, 63, 64, 80
  summary, 9
  summary.globaldiag, 6, 21, 37, 67, 68
  summary.lista, 69
<table>
<thead>
<tr>
<th>Function</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>summary.localdiag</code></td>
<td>8, 10, 24, 39, 69</td>
</tr>
<tr>
<td><code>summary.localtest</code></td>
<td>18, 25, 40, 71</td>
</tr>
<tr>
<td><code>summary.locstppm</code></td>
<td>26, 41, 72</td>
</tr>
<tr>
<td><code>summary.sepstlppm</code></td>
<td>73</td>
</tr>
<tr>
<td><code>summary.sepstppm</code></td>
<td>74</td>
</tr>
<tr>
<td><code>summary.stcov</code></td>
<td>75</td>
</tr>
<tr>
<td><code>summary.stlgcppm</code></td>
<td>30, 62, 75</td>
</tr>
<tr>
<td><code>summary.stlp</code></td>
<td>31, 46, 63, 76</td>
</tr>
<tr>
<td><code>summary.stlpm</code></td>
<td>77</td>
</tr>
<tr>
<td><code>summary.stp</code></td>
<td>33, 47, 63, 78</td>
</tr>
<tr>
<td><code>summary.stpm</code></td>
<td>79</td>
</tr>
<tr>
<td><code>summary.stppm</code></td>
<td>36, 66, 80</td>
</tr>
<tr>
<td><code>valenciacrimes</code></td>
<td>81</td>
</tr>
<tr>
<td><code>valencianet</code></td>
<td>82</td>
</tr>
</tbody>
</table>