

Package ‘ppdiag’

August 12, 2021

Type Package

Title Diagnosis and Visualizations Tools for Temporal Point Processes

Version 0.1.1

Description A suite of diagnostic tools for univariate point processes.

This includes tools for simulating and fitting both common and more complex temporal point processes. We also include functions to visualise these point processes and collect existing diagnostic tools of Brown et al. (2002) <[doi:10.1162/08997660252741149](https://doi.org/10.1162/08997660252741149)> and Wu et al. (2021) <[doi:10.1002/9781119821588.ch7](https://doi.org/10.1002/9781119821588.ch7)>, which can be used to assess the fit of a chosen point process model.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.1

URL <https://owenward.github.io/ppdiag/>

BugReports <https://github.com/OwenWard/ppdiag/issues>

Depends R (>= 3.5)

Imports graphics, stats

Suggests testthat (>= 2.1.0), knitr, rmarkdown, covr

VignetteBuilder knitr

NeedsCompilation no

Author Sally Sun [aut],
Owen G. Ward [aut, cre] (<<https://orcid.org/0000-0002-9678-3542>>),
Jing Wu [aut],
Xiaoxi Zhao [aut],
Lihao Xiao [ctb],
Tian Zheng [aut]

Maintainer Owen G. Ward <owen.ward@columbia.edu>

Repository CRAN

Date/Publication 2021-08-12 13:50:02 UTC

R topics documented:

| | |
|----------------------|-----------|
| drawHPIntensity | 2 |
| drawHPPIIntensity | 3 |
| drawUniMMHPIntensity | 4 |
| drawUniMMPPIntensity | 5 |
| fithp | 6 |
| fithpp | 7 |
| intensityqqplot | 7 |
| pp_compensator | 8 |
| pp_diag | 9 |
| pp_hp | 10 |
| pp_hpp | 11 |
| pp_ksplot | 11 |
| pp_mmhp | 12 |
| pp_mmpp | 13 |
| pp_qqexp | 13 |
| pp_residual | 14 |
| pp_simulate | 15 |
| Index | 16 |

| | |
|-----------------|---|
| drawHPIntensity | <i>Draw the intensity of Hawkes Process</i> |
|-----------------|---|

Description

Draw the intensity of a Hawkes Process

Usage

```
drawHPIntensity(
  hp = NULL,
  events,
  int_title = "Hawkes Intensity",
  start = 0,
  end = max(events),
  history = NULL,
  color = 1,
  i = 1,
  add = FALSE,
  fit = FALSE,
  plot_events = TRUE,
  verbose = FALSE
)
```

Arguments

| | |
|-------------|--|
| hp | object parameters for Hawkes process. |
| events | the event times happened in this state |
| int_title | title of the intensity plot |
| start | the start time of current state |
| end | the end time of current state |
| history | the past event times |
| color | specify the default plotting color. |
| i | state number, used only for drawUniMMHPIntensity |
| add | whether to add the hawkes intensity to an existing plot, used for drawUniMMHPIntensity |
| fit | a boolean indicating whether to fit a new HP to events |
| plot_events | indicate whether events will be plotted |
| verbose | whether to output informative messages as running |

Value

no return value, intensity plot of Hawkes process

Examples

```
set.seed(100)
hp_obj <- pp_hp(lambda0 = 0.5, alpha = 0.45, beta = 0.5)
events <- pp_simulate(hp_obj, start = 0, end = 20)
drawHPPIntensity(hp_obj, events)
```

| | |
|------------------|--|
| drawHPPIntensity | <i>Draw intensity of homogeneous Poisson process</i> |
|------------------|--|

Description

Draw the intensity for a homogeneous Poisson process

Usage

```
drawHPPIntensity(
  hpp = NULL,
  events,
  int_title = "Homogeneous Poisson Process",
  start = 0,
  end = max(events),
  color = "red",
  plot_events = TRUE,
  fit = FALSE,
  add = FALSE,
  verbose = FALSE
)
```

Arguments

| | |
|-------------|--|
| hpp | object for homogeneous Poisson process |
| events | event times input |
| int_title | the plot title |
| start | start of events |
| end | end of events |
| color | a specification for the default plotting color. |
| plot_events | a boolean indicating whether input events will be plotted |
| fit | a boolean indicating whether to fit a hpp or use the passed object |
| add | whether to add the hpp intensity to an existing plot |
| verbose | whether to output informative messages as running |

Value

no return value, intensity plot of homogeneous Poisson process

Examples

```
pois_y <- pp_hpp(lambda = 1)
drawHPPIntensity(pois_y, events = pp_simulate(pois_y, end = 10))
```

drawUniMMHPIntensity *Draw the intensity of the Markov-modulated Hawkes Process(MMHP)*

Description

Take a mmhp object and draw its intensity accordingly

Usage

```
drawUniMMHPIntensity(
  mmhp,
  simulation,
  int_title = "Intensity of MMHP",
  leg_location = "topright",
  color = 1,
  add = FALSE
)
```

Arguments

| | |
|--------------|---|
| mmhp | a mmhp object including its state, state_time, events, lambda0, lambda1, beta and alpha. |
| simulation | the simulated Markov-modulated Hawkes Process(MMHP) |
| int_title | title of the plot. |
| leg_location | location of legend, if moving needed |
| color | A specification for the default plotting color. |
| add | logical; if TRUE add to an already existing plot; if NA start a new plot taking the defaults for the limits and log-scaling of the x-axis from the previous plot. Taken as FALSE (with a warning if a different value is supplied) if no graphics device is open. |

Value

no return value, intensity plot of Markov-modulated Hawkes process

Examples

```
Q <- matrix(c(-0.4, 0.4, 0.2, -0.2), ncol = 2, byrow = TRUE)
x <- pp_mmhp(Q,
  delta = c(1 / 3, 2 / 3), lambda0 = 0.9, lambda1 = 1.1,
  alpha = 0.8, beta = 1.2
)
y <- pp_simulate(x, n = 25)
drawUniMMHPIntensity(x, y)
```

drawUniMMPPIntensity *Draw the intensity of the Markov-modulated Poisson Process(MMPP)*

Description

Take a mmpp object and draw its intensity accordingly

Usage

```
drawUniMMPPIntensity(
  mmpp,
  simulation,
  add = FALSE,
  color = 1,
  fit = FALSE,
  int_title = "Intensity Plot of MMPP"
)
```

Arguments

| | |
|-------------------------|---|
| <code>mmp</code> | a mmp object including its transition probability matrix, λ_0 , δ , and c . |
| <code>simulation</code> | the simulated Markov-modulated Poisson Process(MMPP) |
| <code>add</code> | logical; if TRUE add to an already existing plot; if NA start a new plot taking the defaults for the limits and log-scaling of the x-axis from the previous plot. Taken as FALSE (with a warning if a different value is supplied) if no graphics device is open. |
| <code>color</code> | A specification for the default plotting color. |
| <code>fit</code> | a boolean indicating whether to fit the events provided |
| <code>int_title</code> | title of the plot. |

Value

no return value, intensity plot of Markov-modulated Poisson process

Examples

```
Q <- matrix(c(-0.4, 0.4, 0.2, -0.2), ncol = 2, byrow = TRUE)
x <- pp_mmp(Q, delta = c(1 / 3, 2 / 3), lambda0 = 0.9, c = 1.2)
y <- pp_simulate(x, n = 10)
drawUniMMPPIIntensity(x, y)
```

fithp

Determine the MLE of Hawkes process numerically

Description

Determine the MLE of Hawkes process numerically

Usage

```
fithp(events, end = max(events), vec = c(0.1, 0.2, 0.3))
```

Arguments

| | |
|---------------------|--|
| <code>events</code> | event times |
| <code>end</code> | end of observation period starting from 0 (default last event) |
| <code>vec</code> | vector of initial parameter values |

Value

a hp object indicating the maximum likelihood parameter values (λ_0, α, β) for Hawkes process. This is a non-convex problem and a (unique) solution is not guaranteed.

Examples

```
hp_obj <- pp_hp(lambda0 = 0.1, alpha = 0.45, beta = 0.5)
sims <- pp_simulate(hp_obj, start = 0, n = 10)
fithp(sims)
```

fithpp*Fit a homogeneous poisson process to event data*

Description

Compute maximum likelihood estimator of the rate of a homogeneous Poisson process for the given events.

Usage

```
fithpp(events, end = max(events))
```

Arguments

events vector containing the event times.
end end of observation period, starting from 0 (default is last event)

Value

a hpp object containing the events and the estimated parameter

Examples

```
pois_y <- pp_hpp(lambda = 1)
events <- pp_simulate(pois_y, end = 10)
fithpp(events)
```

intensityqqplot*Draw intensity of fitted point process and QQ-Plot of rescaled events*

Description

Draw the intensity and q-q plot for models

Usage

```

intensityqqplot(object, events, markov_states)

## Default S3 method:
intensityqqplot(object, events, markov_states)

## S3 method for class 'hp'
intensityqqplot(object, events, markov_states = NULL)

## S3 method for class 'hpp'
intensityqqplot(object, events, markov_states = NULL)

## S3 method for class 'mmp'
intensityqqplot(object, events = markov_states$events, markov_states)

## S3 method for class 'mmhp'
intensityqqplot(object, events = markov_states$events, markov_states)

```

Arguments

object parameters for the models: hp, hpp, and mmhp
events event times
markov_states only for mmp and mmhp, markov states simulation output

Value

no return value, intensity and qq-plot in a single plot

pp_compensator *Compensators for point processes*

Description

Computes the compensator for included point processes

Usage

```

pp_compensator(object, events)

## Default S3 method:
pp_compensator(object, events)

## S3 method for class 'mmp'
pp_compensator(object, events)

## S3 method for class 'hp'

```



```
pp_compensator(object, events)

## S3 method for class 'mmhp'
pp_compensator(object, events)

## S3 method for class 'hpp'
pp_compensator(object, events)
```

Arguments

| | |
|--------|--|
| object | a point process model |
| events | event times, which can have first value as 0 |

Value

compensator vector of rescaled interevent times

Examples

```
hpp_obj <- pp_hpp(lambda = 1)
events <- pp_simulate(hpp_obj, end = 10)
comp <- pp_compensator(hpp_obj, events)
```

pp_diag

Summarise diagnostics for point process models

Description

Generate diagnostic tools for different point process models, including quantile-quantile plot, ks plot, raw residual and pearson residual.

Usage

```
pp_diag(object, events)

## Default S3 method:
pp_diag(object, events)

## S3 method for class 'hp'
pp_diag(object, events)

## S3 method for class 'mmhp'
pp_diag(object, events)

## S3 method for class 'mmp'
pp_diag(object, events)

## S3 method for class 'hpp'
pp_diag(object, events)
```

Arguments

| | |
|--------|-----------------------|
| object | a point process model |
| events | event times |

Value

Invisibly returns NULL. Outputs plots and summary of diagnostics to console

Examples

```
hpp_obj <- pp_hpp(lambda = 1)
events <- pp_simulate(hpp_obj, end = 50)
pp_diag(hpp_obj, events)
```

| | |
|-------|---------------------------------------|
| pp_hp | <i>Create a Hawkes process object</i> |
|-------|---------------------------------------|

Description

Create a Hawkes Process with an exponential kernel according to the given parameters: lambda0, alpha, beta and events. If events are missing, then it means that data will be added later(simulated from this process)

Usage

```
pp_hp(lambda0, alpha, beta, events = NULL)
```

Arguments

| | |
|---------|---|
| lambda0 | initial intensity at the start time |
| alpha | jump size in increase of intensity |
| beta | exponential decay of intensity |
| events | vector containing the event times. Note that the first event is at time zero. Alternatively, events could be specified as NULL, meaning that the data will be added later (e.g. simulated). |

Value

hp object

Examples

```
pp_hp(lambda0 = 0.1, alpha = 0.45, beta = 0.5)
```

`pp_hpp`*Create a homogeneous Poisson process object*

Description

Create a homogeneous Poisson object according to given parameters: lambda, and events. If events are missing, then it means that data will be added later(simulated from this process).

Usage

```
pp_hpp(lambda, events = NULL)
```

Arguments

| | |
|--------|-----------------------------|
| lambda | rate of the Poisson process |
| events | event times, optional |

Value

hpp object

Examples

```
pp_hpp(lambda = 1)
```

`pp_kspplot`*KS plot of empirical and theoretical cdf curve of fitted point process*

Description

Plot empirical cdf plot for rescaled-inter-event-times and exponential cdf as a reference curve

Usage

```
pp_kspplot(r, ...)
```

Arguments

| | |
|-----|----------------------------|
| r | rescaled-inter-event-times |
| ... | other arguments for plots |

Value

no return value, KS plot for rescaled-inter-event-times and exponential cdf curve

pp_mmhp

*Create a Markov-modulated Hawkes Process(MMHP) object***Description**

Create a Markov-modulated Hawkes Process(MMHP) model according to the given parameters: lambda0, lambda1, alpha, beta, event times and transition probability matrix. If event time events is missing, then it means that data will be added later(e.g. simulated)

Usage

```
pp_mmhp(lambda0, lambda1, alpha, beta, Q = NULL, delta = NULL, events = NULL)
```

Arguments

| | |
|---------|---|
| lambda0 | intensity for homogeneous Poisson process. |
| lambda1 | base intensity for Hawkes process. |
| alpha | jump size of the increase in intensity in the hawkes process |
| beta | exponential decrease of intensity in the hawkes process |
| Q | transition probability matrix. |
| delta | initial state probability. |
| events | vector containing the event times. Note that the first event is at time zero. Alternatively, events could be specified as NULL, meaning that the data will be added later (e.g. simulated). |

Value

mmhp object

Examples

```
Q <- matrix(c(-0.4, 0.4, 0.2, -0.2), ncol = 2, byrow = TRUE)
pp_mmhp(Q,
  delta = c(1 / 3, 2 / 3), lambda0 = 0.9, lambda1 = 1.1,
  alpha = 0.8, beta = 1.2
)
```

pp_mmpp *Create a Markov-modulated Poisson Process(MMPP) object*

Description

Create a Markov-modulated Poisson Process(MMPP) model according to the given parameters: lambda0, c, q1, q2 and event times. If event time tau is missing, then it means that data will be added later(e.g. simulated)

Usage

```
pp_mmpp(lambda0, c, Q, events = NULL, delta = NULL)
```

Arguments

| | |
|---------|--|
| lambda0 | parameters for Poisson process. |
| c | the proportion of intensity 1 over intensity 2 |
| Q | transition probability matrix |
| events | vector containing the event times. Note that the first event is often specified as zero. Alternatively, events could be specified as NULL, meaning that the data will be added later (e.g. simulated). |
| delta | initial state probability. |

Value

mmpp object

Examples

```
Q <- matrix(c(-0.4, 0.4, 0.2, -0.2), ncol = 2, byrow = TRUE)
pp_mmpp(Q = Q, lambda0 = 1, c = 1.5, delta = c(1 / 3, 2 / 3))
```

pp_qqexp *Plot QQ-plot for rescaled-inter-event-times of fitted point process*

Description

Generate Quantile-quantile plot for rescaled-inter-event-times, which are independently and identically distributed as exponential random variables with rate 1 under the true point process.

Usage

```
pp_qqexp(r, ...)
```

Arguments

r rescaled-inter-event-times
 ... other arguments for plots

Value

no return value, quantile-quantile plot for rescaled-inter-event-times

pp_residual *Compute raw and pearson residuals for point process models*

Description

Compute raw and pearson residuals for point process models

Usage

```
pp_residual(object, events, start = 0, end = max(events), steps = 1000)
```

Arguments

object point process model containing the parameters
 events vector of event times
 start start of observation period (default 0)
 end end of observation period (default final event)
 steps number of steps for numeric integration (if needed)

Value

the raw and pearson residuals

Examples

```
Q <- matrix(c(-0.4, 0.4, 0.2, -0.2), ncol = 2, byrow = TRUE)
x <- pp_mmhp(Q,
  delta = c(1 / 3, 2 / 3), lambda0 = 0.9,
  lambda1 = 1.1, alpha = 0.8, beta = 1.2
)
y <- pp_simulate(x, n = 10)
pp_residual(x, events = y$events)
```

pp_simulate

*Simulate events from a temporal point process***Description**

Currently available point processes are homogeneous Poisson, Hawkes with exponential kernel, MMHP and MMPP

Usage

```
pp_simulate(object, start = 0, end = 1, n = NULL, verbose = FALSE)

## Default S3 method:
pp_simulate(object, start = 0, end = 1, n = NULL, verbose = FALSE)

## S3 method for class 'hpp'
pp_simulate(object, start = 0, end = 1, n = NULL, verbose = FALSE)

## S3 method for class 'hp'
pp_simulate(object, start = 0, end = 1, n = NULL, verbose = FALSE)

## S3 method for class 'mmp'
pp_simulate(object, start = 0, end = 1, n = NULL, verbose = FALSE)

## S3 method for class 'mmhp'
pp_simulate(object, start = 0, end = 1, n = NULL, verbose = FALSE)
```

Arguments

| | |
|---------|--|
| object | point process model object of type hpp, hp, mmhp, or mmp |
| start | start time of events simulated. Not used for Markov modulated models |
| end | end time of events simulated. Not used for Markov modulated models |
| n | number of events simulated. Required for Markov modulated models, optional otherwise |
| verbose | whether to output informative messages as running |

Value

a vector of event times for all models. For Markov modulated models, also returns details on the underlying latent process

Examples

```
hpp_obj <- pp_hpp(lambda = 1)
s <- pp_simulate(hpp_obj, n = 50)
```

Index

[drawHPIntensity](#), [2](#)
[drawHPPIntensity](#), [3](#)
[drawUniMMHPIntensity](#), [4](#)
[drawUniMMPPIntensity](#), [5](#)

[fithp](#), [6](#)
[fithpp](#), [7](#)

[intensityqqplot](#), [7](#)

[pp_compensator](#), [8](#)
[pp_diag](#), [9](#)
[pp_hp](#), [10](#)
[pp_hpp](#), [11](#)
[pp_ksplot](#), [11](#)
[pp_mmhp](#), [12](#)
[pp_mmpp](#), [13](#)
[pp_qqexp](#), [13](#)
[pp_residual](#), [14](#)
[pp_simulate](#), [15](#)