

Package ‘poisson’

October 14, 2022

Type Package

Title Simulating Homogenous & Non-Homogenous Poisson Processes

Version 1.0

Date 2015-10-01

Author Kristian Brock [aut] <kristian.brock@gmail.com>, Daniel Slade [ctb]

Maintainer Daniel Slade <sladeD@bham.ac.uk>

Depends methods

Description

Contains functions and classes for simulating, plotting and analysing homogenous and non-homogenous Poisson processes.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-23 16:23:14

R topics documented:

poisson-package	2
hpp.event.times	3
hpp.lik	4
hpp.mean	5
hpp.mean.event.times	6
hpp.mle	6
hpp.plot	7
hpp.scenario	8
hpp.sim	9
nhpp.event.times	10
nhpp.lik	11
nhpp.mean	12
nhpp.mean.event.times	13
nhpp.mle	14
nhpp.plot	15

nhpp.scenario	16
nhpp.sim	17
nhpp.sim.slow	18
plot-methods	19
plotprocesses	20
PoissonProcessScenario-class	21
show-methods	22

Index	23
--------------	-----------

poisson-package	<i>Simulating Homogenous & Non-Homogenous Poisson Processes</i>
-----------------	---

Description

This package contains a functions and classes for simulating, plotting and analysing homogenous and non-homogenous Poisson processes.

Details

Package: poisson
 Type: Package
 Version: 1.0
 Date: 2015-10-01
 License: GPL-2

The original motivation for this package was modelling recruitment to clinical trials. The gap between patients registering is random. There were examples where we expected that gap to be the same, on average, throughout the trial and for this problem, we simulated patient arrival times as homogeneous Poisson processes. In multi-centre trials, however, we expected that gap to be large at the start of the trial but get smaller as more recruitment centres opened. This scenario required non-homogeneous Poisson processes. Though this package appeared through a medical statistics application, the ability to simulate and analyse Poisson processes is helpful in lots of fields.

The most useful methods are those that simulate scenarios. A scenario consists of many simulated processes, a mean process, and quantile processes. The mean process shows the average number of events through time, i.e. the most likely process path. The simulated paths and the quantile processes inform the analyst about the level of variance about this mean, allowing inference on best and worst outcomes, as well as the most likely outcome.

Imagine a scenario where we expect 5 events per unit of time, on average, and don't expect that average to change. We want to analyse the distribution of paths and hitting times of observing 20 events. To simulate and view the scenario, run:

```
scen = hpp.scenario(rate = 5, num.events = 20, num.sims = 100)
plot(scen, main='My HPP Scenario')
```

The mean process values are in scen@x.bar and the quantile processes are in scen@x.q.

In contrast, let us now assume that the rate of events will be time-varying. Say we expect the event intensity to start at zero and increase linearly to 100% after three units of time. When event intensity is at 100%, we expect 10 events per unit time. To simulate this scenario, we run:

```
intensity <- function(t) pmin(t/3, 1)
rate <- 10
num.events <- 100
scen = nhpp.scenario(rate, num.events, num.sims = 100, prob.func=intensity)
plot(scen, main='My NHPP Scenario')
```

Author(s)

Kristian Brock [aut], Daniel Slade [ctb]
Maintainer: Daniel Slade <sladeD@bham.ac.uk>

hpp.event.times

Simulate homogeneous Poisson process event times

Description

Randomly sample the num.events consecutive event times of a random homogeneous poisson process with given rate. Note: the rate parameter is often referred to as lambda.

Usage

```
hpp.event.times(rate, num.events, num.sims = 1, t0 = 0)
```

Arguments

rate	The rate at which events occur in the Poisson process, aka lambda
num.events	Number of event times to simulate in each process
num.sims	Number of simulated paths to create
t0	start time

Value

A numeric vector of length num.events if num.sims=1, else, a num.events by num.sims matrix

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.mean.event.times](#), [hpp.scenario](#), [nhpp.event.times](#)

Examples

```
rate <- 10
target <- 50
hpp.event.times(rate, target)
```

`hpp.lik`*Homogeneous Poisson process likelihood*

Description

Get the likelihood of a rate parameter at a specific time for observed HPP event times.

Usage

```
hpp.lik(x, T1, rate)
```

Arguments

<code>x</code>	a vector of HPP event times
<code>T1</code>	Calculate likelihood at this time
<code>rate</code>	the putative HPP event rate

Value

Returns a numerical value for the likelihood.

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.mle](#), [nhpp.lik](#)

Examples

```
X = c(0.17, 0.39, 0.63, 0.78, 0.99)
hpp.lik(X, T1 = 1, rate = 4)
hpp.lik(X, T1 = 1, rate = 5)
hpp.lik(X, T1 = 1, rate = 6)
# 5 is the most likely of these three rates
```

hpp.mean	<i>Expected value of an homogeneous Poisson process.</i>
----------	--

Description

Calculate the expected value of an homogeneous Poisson process at regular points in time.

Usage

```
hpp.mean(rate, t0 = 0, t1 = 1, num.points = 100, maximum = NULL)
```

Arguments

rate	The rate at which events occur in the Poisson process, aka lambda
t0	Start time
t1	End time
num.points	Number of points to use between t0 and t1 in calculating the mean
maximum	The optional maximum value that the process should take

Value

A numeric vector of length num.points

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.scenario](#), [nhpp.mean](#)

Examples

```
hpp.mean(rate = 20, t1 = 5, maximum = 50)
```

hpp.mean.event.times *Expected event times of an homogeneous Poisson process.*

Description

Calculate the expected event times of an homogeneous Poisson process.

Usage

```
hpp.mean.event.times(rate, num.events)
```

Arguments

rate	The rate at which events occur in the Poisson process, aka lambda
num.events	Observe mean event times at this many points

Value

A vector of length num.events giving the expected times

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.scenario](#), [nhpp.mean.event.times](#)

Examples

```
rate <- 10  
hpp.mean.event.times(rate, 50)
```

hpp.mle *Get the maximum-likelihood rate parameter of an HPP (homogenous Poisson process)*

Description

Get the maximum-likelihood rate parameter for given HPP event times.

Usage

```
hpp.mle(x, T1)
```

Arguments

`x` a vector of HPP event times
`T1` Calculate MLE at this time

Value

Returns a numeric value, the maximum-likelihood rate parameter

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.lik](#), [nhpp.mle](#)

Examples

```
X = c(0.17, 0.39, 0.63, 0.78, 0.99)
hpp.mle(X, T1=1)
```

 hpp.plot

Plot simulated homogeneous Poisson processes

Description

Simulate and plot simulated homogeneous Poisson processes, also returning the mean and quantile processes.

Usage

```
hpp.plot(rate, num.events, num.sims = 100, t0 = 0, t1 = NULL,
         num.points = 100, quantiles = c(0.025, 0.975), ...)
```

Arguments

`rate` The rate at which events occur in the Poisson process, aka lambda
`num.events` Number of event times to simulate in each process
`num.sims` Number of simulated paths to plot
`t0` Start time
`t1` End time
`num.points` Number of points to use in estimating mean and quantile processes
`quantiles` plot these quantile processes
`...` further arguments to be passed to methods

Value

list	
x	Matrix of event times, one process per column
x.bar	Vector of mean process event times
x.q	Matrix of quantile event times, one process per column

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.scenario](#), [nhpp.plot](#)

Examples

```
hpp.plot(rate = 5, num.events = 20, num.sims = 100, main='My simulated HPPs')
```

hpp.scenario

Simulate an homogeneous Poisson process scenario

Description

Simulate an homogeneous Poisson process scenario, with sample paths, expected value process, and quantile processes.

Usage

```
hpp.scenario(rate, num.events, num.sims = 100, t0 = 0, t1 = NULL,
             num.points = 100, quantiles = c(0.025, 0.975), ...)
```

Arguments

rate	The rate at which events occur in the Poisson process, aka lambda
num.events	Number of event times to simulate in each process
num.sims	Number of simulated paths to plot
t0	Start time
t1	End time
num.points	Number of points to use in estimating mean and quantile processes
quantiles	plot these quantile processes
...	further arguments to be passed to or from methods

Value

Instance of PoissonProcessScenario

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[nhpp.scenario](#), [PoissonProcessScenario](#)

Examples

```
scen = hpp.scenario(rate = 5, num.events = 20, num.sims = 100)
scen@x.bar
plot(scen, main='My HPP Scenario')
```

hpp.sim

Simulate homogeneous Poisson process(es).

Description

Get the n consecutive event times of an homogeneous poisson process with given rate. Note: the rate parameter is often referred to as lambda.

Usage

```
hpp.sim(rate, num.events, num.sims = 1, t0 = 0, prepend.t0 = T)
```

Arguments

rate	The rate at which events occur in the Poisson process, aka lambda
num.events	Number of event times to simulate in each process
num.sims	Number of simulated paths to create
t0	Start time
prepend.t0	TRUE: To include t0 at the start of the process, FALSE: Not include to at the start of the process.

Value

A numeric vector of length num.events if num.sims=1, else, a num.events by num.sims matrix [num.events+1 if prepend.zero=T]

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.scenario](#), [nhpp.sim](#)

Examples

```
rate <- 10
target = 50
hpp.sim(rate, target)
```

nhpp.event.times	<i>Simulate non-homogeneous Poisson process event times</i>
------------------	---

Description

Randomly simulate the num.events consecutive event times of a non-homogeneous poisson process. Events are simulated using an underlying homogeneous process with given rate. An event at time t is admitted with probability prob.func(t). Note: The rate parameter of an homogeneous process is often called lambda.

Usage

```
nhpp.event.times(rate, num.events, prob.func, num.sims = 1, t0 = 0)
```

Arguments

rate	the rate at which events occur in the equivalent homogeneous Poisson process, aka lambda
num.events	number of event times to simulate in each process
prob.func	aka intensity function, function that takes time as sole argument and returns value between 0 and 1
num.sims	number of simulated paths to create
t0	the reference start time of all events

Details

This method is called 'thinning' by Lewis & Shedler (1978).

Value

A numeric vector of length num.events if num.sims=1, else, a num.events by num.sims matrix

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

References

Lewis & Shedler, Simulation of Non-Homogeneous Poisson Processes by Thinning, 1978

See Also

[nhpp.mean.event.times](#), [nhpp.scenario](#), [hpp.event.times](#)

Examples

```
rate <- 10
target <- 50
intensity <- function(t) pmin(t/3, 1)
nhpp.event.times(rate, target, intensity)
```

nhpp.lik

Non-homogeneous Poisson process likelihood

Description

Get the likelihood of a rate parameter at a specific time for observed NHPP event times and given intensity function.

Usage

```
nhpp.lik(x, T1, rate, prob.func)
```

Arguments

x	a vector of HPP event times
T1	Calculate likelihood at this time
rate	the putative HPP event rate
prob.func	aka intensity function, function that takes time as sole argument and returns value between 0 and 1

Value

Returns a numerical value for the likelihood.

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[nhpp.mle](#), [hpp.lik](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
X = c(0.74, 1.50, 1.67, 2.01, 2.27)
nhpp.lik(X, T1 = 2.3, rate = 5, prob.func = intensity)
nhpp.lik(X, T1 = 2.3, rate = 6, prob.func = intensity)
nhpp.lik(X, T1 = 2.3, rate = 7, prob.func = intensity)
# 6 is the most likely of these three rates
```

 nhpp.mean

Expected value of a non-homogeneous Poisson process.

Description

Calculate the expected value of a non-homogeneous Poisson process at points in time.

Usage

```
nhpp.mean(rate, prob.func, t0 = 0, t1 = 1, num.points = 100, maximum = NULL)
```

Arguments

rate	the rate at which events occur in the Poisson process, aka lambda
prob.func	function that takes time as sole argument and returns value between 0 and 1
t0	start time
t1	end time
num.points	number of points between t0 and t1 to use in estimating mean
maximum	the optional maximum value that the process should take

Value

A numeric vector of length num.points

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[nhpp.scenario](#), [hpp.mean](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
nhpp.mean(rate = 20, t1 = 5, maximum = 50, prob.func=intensity)
```

nhpp.mean.event.times *Expected event times of a non-homogeneous Poisson process.*

Description

Calculate the expected event times of a non-homogeneous Poisson process.

Usage

```
nhpp.mean.event.times(rate, num.events, prob.func, max.time = 1000)
```

Arguments

rate	The rate at which events occur in the Poisson process, aka lambda
num.events	Observe mean event times at this many points
prob.func	aka intensity function, function that takes time as sole argument and returns value between 0 and 1
max.time	Maximum time value to use

Value

A vector of length num.events giving the expected times

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[nhpp.event.times](#), [nhpp.scenario](#), [hpp.mean.event.times](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
rate <- 10
nhpp.mean.event.times(rate, 50, prob.func = intensity)
```

nhpp.mle	<i>Get the maximum-likelihood rate parameter of an NHPP (non-homogenous Poisson process)</i>
----------	--

Description

Get the maximum-likelihood rate parameter for given NHPP event times.

Usage

```
nhpp.mle(x, T1, prob.func, max.val)
```

Arguments

x	a vector of NHPP event times
T1	calculate MLE at this time
prob.func	function that takes time as sole argument and returns value between 0 and 1
max.val	maximum value to consider for MLE of NHPP rate parameter

Value

Returns a numeric value, the maximum-likelihood rate parameter

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[nhpp.lik](#), [hpp.mle](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
X = c(0.74, 1.50, 1.67, 2.01, 2.27)
nhpp.mle(X, T1=1, prob.func=intensity, max.val = 70)
```

nhpp.plot

*Plot simulated non-homogeneous Poisson processes***Description**

Plot num.events simulated non-homogeneous Poisson processes, plus the mean and quantiles

Usage

```
nhpp.plot(rate, num.events, prob.func, num.sims = 100, t0 = 0, t1 = NULL,
          num.points = 100, quantiles = c(0.025, 0.975), ...)
```

Arguments

rate	the rate at which events occur in the Poisson process, aka lambda
num.events	the number of event times to simulate in each process
prob.func	function that takes time as sole argument and returns value between 0 and 1
num.sims	number of simulated paths to plot
t0	start time
t1	end time
num.points	number of points to use in estimating mean and quantile processes
quantiles	plot these quantile processes
...	further arguments to be passed to or from methods

Value

list	
x	Matrix of event times, one process per column
x.bar	Vector of mean process event times
x.q	Matrix of quantile event times, one process per column

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[nhpp.scenario](#), [hpp.plot](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
nhpp.plot(rate = 5, num.events = 20, num.sims = 100, main='My simulated NHPPs',
          prob.func=intensity)
```

nhpp.scenario *Simulate a non-homogeneous Poisson process scenario*

Description

Simulate a non-homogeneous Poisson process scenario, with sample paths, expected value process, and quantile processes.

Usage

```
nhpp.scenario(rate, num.events, prob.func, num.sims = 100, t0 = 0, t1 = NULL,
              num.points = 100, quantiles = c(0.025, 0.975), ...)
```

Arguments

rate	the rate at which events occur in the equivalent homogeneous Poisson process, aka lambda
num.events	Number of event times to simulate in each process
prob.func	aka intensity function, function that takes time as sole argument and returns value between 0 and 1
num.sims	Number of simulated paths to plot
t0	Start time
t1	End time
num.points	Number of points to use in estimating mean and quantile processes
quantiles	plot these quantile processes
...	further arguments to be passed to or from methods

Value

Instance of PoissonProcessScenario

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.scenario](#), [PoissonProcessScenario](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
rate <- 10
num.events <- 100
scen = nhpp.scenario(rate, num.events, num.sims = 100, prob.func=intensity)
scen@x.bar
plot(scen, main='My NHPP Scenario')
```

nhpp.sim	<i>Simulate non-homogeneous Poisson process(es)</i>
----------	---

Description

Get the n consecutive event times of a non-homogeneous poisson process. Events are simulated using an homogeneous process with rate, and an event at time t is admitted with probability $\text{prob.func}(t)$. The rate parameter of an homogeneous process is often called lambda.

Usage

```
nhpp.sim(rate, num.events, prob.func, num.sims = 1, t0 = 0, prepend.t0 = T)
```

Arguments

rate	the rate at which events occur in the equivalent homogeneous Poisson process, aka lambda
num.events	number of event times to simulate in each process
prob.func	aka intensity function, function that takes time as sole argument and returns value between 0 and 1
num.sims	number of simulated paths to create
t0	the reference start time of all events
prepend.t0	T to include t0 at the start of the process

Details

This method is called 'thinning' by Lewis & Shedler (1978)

Value

a numeric vector of length num.events if $\text{num.sims}=1$ else, a num.events by num.sims matrix [num.events+1 is prepend.zero=T]

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

References

Lewis & Shedler, Simulation of Non-Homogeneous Poisson Processes by Thinning, 1978

See Also

[nhpp.scenario](#), [hpp.sim](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
rate <- 10
num.events <- 100
nhpp.sim(rate, num.events, prob.func=intensity)
```

 nhpp.sim.slow

Simulate a non-homogeneous Poisson process.

Description

Get the n consecutive event times of a non-homogeneous poisson process. Events are simulated using an homogeneous process with rate, and an event at time t is admitted with probability $\text{prob.func}(t)$.

Usage

```
nhpp.sim.slow(rate, num.events, prob.func, num.sims = 1, t0 = 0, prepend.t0 = T)
```

Arguments

rate	the rate at which events occur in the equivalent homogeneous Poisson process, aka lambda
num.events	number of event times to simulate in each process
prob.func	aka intensity function, function that takes time as sole argument and returns value between 0 and 1
num.sims	number of simulated paths to create
t0	the reference start time of all events
prepend.t0	T to include t0 at the start of the process

Details

This method is called 'thinning' by Lewis & Shedler (1978)

Value

a numeric vector of length num.events if num.sims=1 else, a num.events by num.sims matrix

Note

This item is my original (slower) implementation of NHPP simulation, hence the name. It does not use recursion so the code is easier to understand.

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

References

Lewis & Shedler, Simulation of Non-Homogeneous Poisson Processes by Thinning, 1978

See Also

[nhpp.scenario](#), [hpp.sim](#)

Examples

```
intensity <- function(t) pmin(t/3, 1)
rate <- 10
num.events <- 100
nhpp.sim.slow(rate, num.events, prob.func=intensity)
```

plot-methods

plot

Description

A simulated scenario can be visualised with a plot. Included are process paths, the mean process and quartile processes.

Usage

```
## S4 method for signature 'PoissonProcessScenario'
plot(x, plot.mean, plot.quantiles, ...)
```

Arguments

x	The PoissonProcessScenario object to plot
plot.mean	TRUE to plot the mean process
plot.quantiles	TRUE to plot the quantile processes
...	Additional arguments affecting the plot

Examples

```
scen = hpp.scenario(rate = 5, num.events = 20, num.sims = 100)
scen@x.bar
plot(scen, main='My HPP Scenario')
```

`plotprocesses`*Plot simulated process paths*

Description

Plot a matrix of simulated process paths

Usage

```
plotprocesses(x, y = NULL, xlab = "t (years)", ylab = "N", type = "l",  
             lty = 2, col = "cadetblue3", xlim = c(0, 1.1 * max(x)),  
             lwd = 0.5, add = F, ...)
```

Arguments

<code>x</code>	matrix of process paths
<code>y</code>	variable for y axis, index of x if NULL
<code>xlab</code>	Label for x-axis
<code>ylab</code>	Label for y-axis
<code>type</code>	Type of plot for simulated processes paths
<code>lty</code>	Line type for simulated processes paths
<code>col</code>	Colour for simulated processes paths
<code>xlim</code>	The range for the x-axis
<code>lwd</code>	Line-width for simulated processes paths
<code>add</code>	TRUE to add to existing plot; FALSE to start afresh
<code>...</code>	Additional arguments affecting the plot

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

Examples

```
scen = hpp.scenario(rate = 5, num.events = 20, num.sims = 100)  
plotprocesses(scen@x, main='My HPP paths')
```

PoissonProcessScenario-class
Class "PoissonProcessScenario"

Description

This class is the result of a Poisson process simulation scenario, yielded by methods like [hpp.scenario](#) and [nhpp.scenario](#). The object has slots for the simulated random processes, the mean process, and quantile processes. It has specific implementations of plot and show.

Objects from the Class

Objects can be created by calls of the form `new("PoissonProcessScenario", ...)`, although they would more commonly be fetched from calls to [hpp.scenario](#) and [nhpp.scenario](#).

Slots

`x`: Object of class "matrix", the simulated process paths
`x.bar`: Object of class "numeric", the mean process
`x.bar.index`: Object of class "numeric", the time variable of the mean process
`x.q`: Object of class "matrix", the quantile processes.

Methods

plot signature(x = "PoissonProcessScenario"): ...
show signature(object = "PoissonProcessScenario"): ...

Author(s)

Kristian Brock - Author, Daniel Slade - Contributor

See Also

[hpp.scenario](#), [nhpp.scenario](#)

show-methods	show
--------------	------

Description

A simulated scenario can be examined with a show.

Usage

```
## S4 method for signature 'PoissonProcessScenario'  
show(object)
```

Arguments

object The [PoissonProcessScenario](#) object to show

Examples

```
scen = hpp.scenario(rate = 5, num.events = 20, num.sims = 100)  
show(scen)
```

Index

- * **HPP**
 - [hpp.event.times, 3](#)
 - [hpp.lik, 4](#)
 - [hpp.mean, 5](#)
 - [hpp.mean.event.times, 6](#)
 - [hpp.mle, 6](#)
 - [hpp.plot, 7](#)
 - [hpp.scenario, 8](#)
 - [hpp.sim, 9](#)
 - [PoissonProcessScenario-class, 21](#)
- * **MLE**
 - [hpp.mle, 6](#)
 - [nhpp.mle, 14](#)
- * **NHPP**
 - [nhpp.event.times, 10](#)
 - [nhpp.lik, 11](#)
 - [nhpp.mean.event.times, 13](#)
 - [nhpp.mle, 14](#)
 - [nhpp.plot, 15](#)
 - [nhpp.scenario, 16](#)
 - [PoissonProcessScenario-class, 21](#)
- * **classes**
 - [PoissonProcessScenario-class, 21](#)
- * **likelihood**
 - [hpp.lik, 4](#)
 - [nhpp.lik, 11](#)
- * **maximum likelihood**
 - [hpp.mle, 6](#)
 - [nhpp.mle, 14](#)
- * **methods**
 - [plot-methods, 19](#)
 - [show-methods, 22](#)
- * **nhpp**
 - [nhpp.mean, 12](#)
 - [nhpp.sim, 17](#)
 - [nhpp.sim.slow, 18](#)
- * **plot**
 - [plot-methods, 19](#)
- * **poisson**
 - [hpp.event.times, 3](#)
 - [hpp.lik, 4](#)
 - [hpp.mean, 5](#)
 - [hpp.mean.event.times, 6](#)
 - [hpp.mle, 6](#)
 - [hpp.plot, 7](#)
 - [hpp.scenario, 8](#)
 - [hpp.sim, 9](#)
 - [nhpp.event.times, 10](#)
 - [nhpp.lik, 11](#)
 - [nhpp.mean, 12](#)
 - [nhpp.mean.event.times, 13](#)
 - [nhpp.mle, 14](#)
 - [nhpp.plot, 15](#)
 - [nhpp.scenario, 16](#)
 - [nhpp.sim, 17](#)
 - [nhpp.sim.slow, 18](#)
 - [plotprocesses, 20](#)
 - [PoissonProcessScenario-class, 21](#)
- * **show**
 - [show-methods, 22](#)
- * **simulation**
 - [hpp.event.times, 3](#)
 - [hpp.scenario, 8](#)
 - [hpp.sim, 9](#)
 - [nhpp.event.times, 10](#)
 - [nhpp.scenario, 16](#)
 - [nhpp.sim, 17](#)
 - [nhpp.sim.slow, 18](#)
- [hpp.event.times, 3, 11](#)
- [hpp.lik, 4, 7, 11](#)
- [hpp.mean, 5, 12](#)
- [hpp.mean.event.times, 3, 6, 13](#)
- [hpp.mle, 4, 6, 14](#)
- [hpp.plot, 7, 15](#)
- [hpp.scenario, 3, 5, 6, 8, 8, 9, 16, 21](#)
- [hpp.sim, 9, 17, 19](#)
- [nhpp.event.times, 3, 10, 13](#)

`nhpp.lik`, [4](#), [11](#), [14](#)
`nhpp.mean`, [5](#), [12](#)
`nhpp.mean.event.times`, [6](#), [11](#), [13](#)
`nhpp.mle`, [7](#), [11](#), [14](#)
`nhpp.plot`, [8](#), [15](#)
`nhpp.scenario`, [9](#), [11–13](#), [15](#), [16](#), [17](#), [19](#), [21](#)
`nhpp.sim`, [9](#), [17](#)
`nhpp.sim.slow`, [18](#)

`plot`, `PoissonProcessScenario`-method
 (plot-methods), [19](#)

plot-methods, [19](#)

plotprocesses, [20](#)

poisson (poisson-package), [2](#)

poisson-package, [2](#)

`PoissonProcessScenario`, [9](#), [16](#), [19](#), [22](#)

`PoissonProcessScenario`-class, [21](#)

`show`, `PoissonProcessScenario`-method
 (show-methods), [22](#)

show-methods, [22](#)