Package: RHawkes (via r-universe)

September 13, 2024

Type Package

Title Renewal Hawkes Process

Version 1.0

Date 2022-5-4

Description The renewal Hawkes (RHawkes) process (Wheatley, Filimonov, and Sornette, 2016 <doi:10.1016/j.csda.2015.08.007>) is an extension to the classical Hawkes self-exciting point process widely used in the modelling of clustered event sequence data. This package provides functions to simulate the RHawkes process with a given immigrant hazard rate function and offspring birth time density function, to compute the exact likelihood of a RHawkes process using the recursive algorithm proposed by Chen and Stindl (2018) <doi:10.1080/10618600.2017.1341324>, to compute the Rosenblatt residuals for goodness-of-fit assessment, and to predict future event times based on observed event times up to a given time. A function implementing the linear time RHawkes process likelihood approximation algorithm proposed in Stindl and Chen (2021) <doi:10.1007/s11222-021-10002-0> is also included.

License GPL (>= 2)

Depends R (>= 2.10), IHSEP

NeedsCompilation no

Author Feng Chen [aut, cre] (<https://orcid.org/0000-0002-9646-3338>), Tom Stindl [ctb] (<https://orcid.org/0000-0001-8627-9337>)

Maintainer Feng Chen <feng.chen@unsw.edu.au>

Date/Publication 2022-05-05 14:40:09 UTC

Repository https://fchenhku.r-universe.dev

RemoteUrl https://github.com/cran/RHawkes

RemoteRef HEAD

RemoteSha 9a033bc7a3c829df6aa494c2bf79501c7152df3d

Contents

RHawkes-package *Renewal Hawkes Process*

Description

The renewal Hawkes (RHawkes) process (Wheatley, Filimonov, and Sornette, 2016 <doi:10.1016/j.csda.2015.08.007>) is an extension to the classical Hawkes self-exciting point process widely used in the modelling of clustered event sequence data. This package provides functions to simulate the RHawkes process with a given immigrant hazard rate function and offspring birth time density function, to compute the exact likelihood of a RHawkes process using the recursive algorithm proposed by Chen and Stindl (2018) <doi:10.1080/10618600.2017.1341324>, to compute the Rosenblatt residuals for goodness-of-fit assessment, and to predict future event times based on observed event times up to a given time. A function implementing the linear time RHawkes process likelihood approximation algorithm proposed in Stindl and Chen (2021) <doi:10.1007/s11222-021-10002-0> is also included.

Details

The DESCRIPTION file:

Package:	RHawkes
Type:	Package
Title:	Renewal Hawkes Process
Version:	1.0
Date:	2022-5-4
Authors@R:	c(person(given="Feng", family="Chen", role = c("aut", "cre"), email = "feng.chen@unsw.edu.au", com
Description:	The renewal Hawkes (RHawkes) process (Wheatley, Filimonov, and Sornette, 2016 <doi:10.1016/j.csda
License:	GPL (>=2)
Depends:	R (>= 2.10), IHSEP

NeedsCompilation:	no
Author:	Feng Chen [aut, cre] (<https://orcid.org/0000-0002-9646-3338>), Tom Stindl [ctb] (<https://orcid.org/00
Maintainer:	Feng Chen <feng.chen@unsw.edu.au>

Index of help topics:

`EM1partial`	`Partial EM algorithm for the RHawkes process, version 1`
`EM2partial`	`Partial EM algorithm for the RHawkes process, version 2`
`RHawkes-package`	`Renewal Hawkes Process`
`damllRH`	`Dynamically approxomated minus loglikelihood of a RHawkes model`
`mllRH`	`Minus loglikelihood of a RHawkes model`
`mllRH1`	`Minus loglikelihood of a RHawkes model with parent probabilities`
`mllRH2`	`Minus loglikelihood of a RHawkes model with Rosenblatt residuals`
`pred.den`	`RHawkes predictive density function`
`pred.haz`	`RHawkes predictive hazard function`
`quake`	`An RHawkes earthquake data set`
`sim.pred`	`Simulate a fitted RHawkes process model`
`sim.pred1`	`Simulate a fitted RHawkes process model for prediction purposes`
`simRHawkes`	`Simulate a renewal Hawkes (RHawkes) process`
`simRHawkes1`	`Simulate a renewal Hawkes (RHawkes) process`
`tms`	`mid-price change times of the AUD/USD exchange rate`

Author(s)

NA

Maintainer: NA

`damllRH` *Dynamically approxomated minus loglikelihood of a RHawkes model*

Description

Calculates an apprximation to the minus loglikelihood of a RHawkes model with given immigration hazard function μ, offspring birth time density function h and branching ratio η relative to event times `tms` on interval $[0, cens]$.

Usage

```
damllRH(tms, cens, par, q=0.999, qe=0.999,
      h.fn = function(x, p) dexp(x, rate = 1 / p),
      mu.fn = function(x, p) {
        exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
       pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
      },
      H.fn = function(x, p) pexp(x, rate = 1 / p),
      Mu.fn = function(x, p) {
       -pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
      },
      keepB=FALSE,
      H.inv=function(x,p)qexp(x,rate=1/p) )
```

Arguments

`tms` A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.

`cens` A numericl scalar. The censoring time.

`par` A numeric vector containing the parameters of the model, in order of the immigration parameters, in $\mu(.)$, offspring distribution parameters, in $h(.)$, and lastly the branching ratio $\eta(.)$.

`q` A numeric scalar in (0,1] and close to 1, which controls how far we look back when truncating the distribution of the most recent immigrant.

`qe` A numeric scalar in (0,1] and close to 1, which controls how to truncation is used in the offspring birth time distribution.

`h.fn` A (vectorized) function. The offspring birth time density function.

`mu.fn` A (vectorized) function. The immigrant waiting time hazard function.

`H.fn` A (vectorized) function. Its value at `t` gives the integral of the offspring birth time density function from 0 to `t`.

`Mu.fn` A (vectorized) function. Its value at `t` gives the integral of the immigrant waiting time hazard function from 0 to `t`.

`keepB` A boolean scalar, indicating whether the looking back values `B_i` should be part of the output or not.

`H.inv` A (vectorized) function, giving the inverse function of the integral of the excitation.

Value

A scalar giving the value of the (approximate) negative log-likelihood, when `keepB` is FALSE (the default); A list with components `mll`, whhich contains the value of the negative log-likelihood, `Bs`, which gives the look-back order of the truncation of the distribution of the last immigrant, and `Bes`, which gives the look-forward order in determining how far into the future the excitation effect is allowed to last.

Author(s)

Feng Chen <feng.chen@unsw.edu.au>

Examples

```
## Not run:
## earthquake times over 96 years
data(quake);
tms <- sort(quake$time);
# add some random noise to the simultaneous occurring event times
tms[213:214] <- tms[213:214] +
                    sort(c(runif(1, -1, 1)/(24*60), runif(1, -1, 1)/(24*60)))


## calculate the minus loglikelihood of an RHawkes with some parameters
## the default hazard function and density functions are Weibull and
## exponential respectively
mllRH(tms, cens = 96*365.25 , par = c(0.5, 20, 1000, 0.5))
damllRH(tms, cens = 96*365.25 , par = c(0.5, 20, 1000, 0.5),q=1,qe=1)
## calculate the MLE for the parameter assuming known parametric forms
## of the immigrant hazard function and offspring density functions.
system.time(est <- optim(c(0.5, 20, 1000, 0.5),
                        mllRH, tms = tms, cens = 96*365.25,
                        mu.fn=function(x,p)p[1]/p[2]*(x/p[2])^(p[1]-1),
                        Mu.fn=function(x,p)(x/p[2])^p[1],
                        control = list(maxit = 5000, trace = TRUE),
                        hessian = TRUE)
            )
system.time(est1 <- optim(c(0.5, 20, 1000, 0.5),
                        function(p){
                            if(any(p<0)||p[4]<0||p[4]>=1)
                                return(Inf);
                            damllRH(tms = tms, cens = 96*365.25,
                                    mu.fn=function(x,p)p[1]/p[2]*(x/p[2])^(p[1]-1),
                                    Mu.fn=function(x,p)(x/p[2])^p[1],
                                    par=p,q=0.999999,qe=0.999999)
                        },
                        control = list(maxit = 5000, trace = TRUE),
                        hessian = TRUE)
            )
## point estimate by MLE
est$par
est1$par
## standard error estimates:
diag(solve(est$hessian))^0.5
diag(solve(est1$hessian))^0.5

## End(Not run)
```

EM1partial *Partial EM algorithm for the RHawkes process, version 1*

Description

Calculates the RHawkes model parameters via a partial Expectation-Maximization (EM1) algorithm of Wheatley, Filimonov and Sornette (2016).

Usage

```
EM1partial(tms, cens, pars, maxiter = 1000, tol = 1e-8,
         h.fn = function(x, p) dexp(x, rate = 1 / p),
         mu.fn = function(x, p){
              exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
          pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
         },
         H.fn = function(x, p) pexp(x, rate = 1 / p),
         logg.fn = function(x, p){
              dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
          pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
              - (x / p[2])^p[1]},
         Mu.fn = function(x, p){
         - pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
         })
```

Arguments

`tms` A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.

`cens` A scalar. The censoring time.

`pars` A numeric vector containing the parameters of the model, in order of the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.

`maxiter` The maximum number of iterations to perform.

`tol` The algorithm stops when the difference between the previous iteration and current iteration parameters sum is less than `tol`.

`h.fn` A (vectorized) function. The offspring density function.

`mu.fn` A (vectorized) function. The immigration hazard function.

`H.fn` A (vectorized) function. Its value at `t` gives the integral of the offspring density function from 0 to `t`.

`Mu.fn` A (vectorized) function. Its value at `t` gives the integral of the immigrant hazard function from 0 to `t`.

`logg.fn` A (vectorized) function. The log of the immigrant distribution function.

Value

iterations	The number of iterations until convergence
diff	The absolute sum of the difference between the final two parameter estimates
pars	The parameter estimates from the EM algorithm

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
## Not run:
## simulated data
tms <- sort(runif(100,0,100))
## the slower version of the EM algorithms on simulated data with default
## immigrant hazard function
## and offspring density
system.time( est1 <- EM1partial(tms, 101, c(2,1,0.5,1)) )

## End(Not run)
```

EM2partial *Partial EM algorithm for the RHawkes process, version 2*

Description

Calculates the RHawkes model parameters via a partial Expectation-Maximization (EM2) algorithm of Wheatley, Filimonov and Sornette (2016).

Usage

```
EM2partial(tms, cens, pars, maxiter = 1000, tol = 1e-8,
          h.fn = function(x, p) dexp(x, rate = 1 / p),
          mu.fn = function(x, p){
               exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
          pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
          },
          H.fn = function(x, p) pexp(x, rate = 1 / p),
          logg.fn = function(x, p){
               dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
           pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
               - (x / p[2])^p[1]},
          Mu.fn = function(x, p){
          - pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
          })
```

Arguments

`tms` A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.

`cens` A scalar. The censoring time.

`pars` A numeric vector containing the parameters of the model, in order of the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.

`maxiter` The maximum number of iterations to perform.

`tol` The algorithm stops when the difference between the previous iteration and current iteration parameters sum is less than `tol`.

`h.fn` A (vectorized) function. The offspring density function.

`mu.fn` A (vectorized) function. The immigration hazard function.

`H.fn` A (vectorized) function. Its value at `t` gives the integral of the offspring density function from 0 to `t`.

`Mu.fn` A (vectorized) function. Its value at `t` gives the integral of the immigrant hazard function from 0 to `t`.

`logg.fn` A (vectorized) function. The log of the immigrant distribution function.

Value

`iterations` The number of iterations until convergence

`diff` The absolute sum of the difference between the final two parameter estimates

`pars` The parameter estimates from the EM algorithm

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
## Not run:
## simulated data
tms <- sort(runif(100,0,100))
## the quicker version on simulated data with default immigrant hazard function
## and offspring density
system.time( est2 <- EM2partial(tms, 101, c(2,1,0.5,1)) )

## End(Not run)
```

mllRH *Minus loglikelihood of a RHawkes model*

Description

Calculates the minus loglikelihood of a RHawkes model with given immigration hazard function μ, offspring density function h and branching ratio η for event times tms on interval $[0, cens]$.

Usage

```
mllRH(tms, cens, par,
      h.fn = function(x, p) dexp(x, rate = 1 / p),
      mu.fn = function(x, p) {
        exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
       pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
      },
      H.fn = function(x, p) pexp(x, rate = 1 / p),
      Mu.fn = function(x, p) {
       -pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
      })
```

Arguments

tms	A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.
cens	A scalar. The censoring time.
par	A numeric vector containing the parameters of the model, in order of the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.
h.fn	A (vectorized) function. The offspring density function.
mu.fn	A (vectorized) function. The immigration hazard function.
H.fn	A (vectorized) function. Its value at t gives the integral of the offspring density function from 0 to t.
Mu.fn	A (vectorized) function. Its value at t gives the integral of the immigrant hazard function from 0 to t.

Value

The value of the negative log-likelihood.

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
## Not run:
## earthquake times over 96 years
data(quake);
tms <- sort(quake$time);
# add some random noise to the simultaneous occurring event times
tms[213:214] <- tms[213:214] +
                    sort(c(runif(1, -1, 1)/(24*60), runif(1, -1, 1)/(24*60)))


## calculate the minus loglikelihood of an RHawkes with some parameters
## the default hazard function and density functions are Weibull and
## exponential respectively
mllRH(tms, cens = 96*365.25 , par = c(0.5, 20, 1000, 0.5))
## calculate the MLE for the parameter assuming known parametric forms
## of the immigrant hazard function and offspring density functions.
system.time(est <- optim(c(0.5, 20, 1000, 0.5),
                        mllRH, tms = tms, cens = 96*365.25,
                        control = list(maxit = 5000, trace = TRUE),
                        hessian = TRUE)
            )
## point estimate by MLE
est$par
## standard error estimates:
diag(solve(est$hessian))^0.5

## End(Not run)
```

mllRH1 *Minus loglikelihood of a RHawkes model with parent probabilities*

Description

Calculates the minus loglikelihood of a RHawkes model with given immigration hazard function μ, offspring density function h and branching ratio η for event times tms on interval $[0, cens]$. The same as mllRH although this version also returns the parent probabilities.

Usage

```
mllRH1(tms, cens, par,
      h.fn = function(x, p) dexp(x, rate = 1/p),
      mu.fn = function(x, p) {
        exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
       pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
        },
        H.fn = function(x, p) pexp(x, rate = 1/p),
        Mu.fn = function(x, p) {
       -pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
        })
```

Arguments

tms — A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.

cens — A scalar. The censoring time.

par — A numeric vector containing the parameters of the model, in order of the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.

h.fn — A (vectorized) function. The offspring density function.

mu.fn — A (vectorized) function. The immigration hazard function.

H.fn — A (vectorized) function. Its value at t gives the integral of the offspring density function from 0 to t.

Mu.fn — A (vectorized) function. Its value at t gives the integral of the immigrant hazard function from 0 to t.

Value

mll — minus log-likelihood

log.p — parent probabilities

n — number of events

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

See Also

mllRH

Examples

```
tmp <- mllRH1(sort(runif(1000,0,1000)), 1001, c(2,1,0.5,1))
for(i in 1:tmp$n)
  cat(exp(tmp$log.p[i*(i - 1)/2 + 1:i]), "\n")
```

mllRH2 *Minus loglikelihood of a RHawkes model with Rosenblatt residuals*

Description

Calculates the minus loglikelihood of a RHawkes model with given immigration hazard function μ, offspring density function h and branching ratio η for event times tms on interval $[0, cens]$. The same as mllRH although this version also returns the Rosenblatt residuals.

Usage

```
mllRH2(tms, cens, par, h.fn = function(x, p) dexp(x, rate = 1/p),
       mu.fn = function(x, p) {
         exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
      pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))},
       H.fn = function(x, p) pexp(x, rate = 1/p),
       Mu.fn = function(x, p) {
       -pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
       })
```

Arguments

`tms` A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.

`cens` A scalar. The censoring time.

`par` A numeric vector containing the parameters of the model, in order of the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.

`h.fn` A (vectorized) function. The offspring density function.

`mu.fn` A (vectorized) function. The immigration hazard function.

`H.fn` A (vectorized) function. Its value at `t` gives the integral of the offspring density function from 0 to `t`.

`Mu.fn` A (vectorized) function. Its value at `t` gives the integral of the immigrant hazard function from 0 to `t`.

Details

Calculate the RHawkes point process Rosenblatt residuals

Value

`mll`	minus log-likelihood
`U`	Rosenblatt residual of observed event time
`n`	number of events

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

See Also

`mllRH`

Examples

```
## Not run:
tmp <- mllRH2(sort(runif(1000,0,1000)),1001,c(2,1,0.5,1))
par(mfrow=c(1,2))
qqunif<-function(dat,...){
  dat<-sort(as.numeric(dat));
  n<-length(dat);
  pvec<-ppoints(n);
  plot(pvec,dat,xlab="Theoretical Quantiles",
       ylab="Sample Quantiles",main="Uniform Q-Q Plot",...)
}
qqunif(tmp$U)
acf(tmp$U)
ks.test(tmp$U,"punif")

## End(Not run)
```

pred.den *RHawkes predictive density function*

Description

Calculates the predictive density of the next observed event time after the censoring time cens based on observations over the interval [0,cens].

Usage

```
pred.den(x, tms, cens, par,
         h.fn = function(x, p) dexp(x, rate = 1 / p),
         mu.fn = function(x, p) {
           exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
        pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))},
           H.fn = function(x, p) pexp(x, rate = 1 / p),
           Mu.fn = function(x, p) {
         -pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
           })
```

Arguments

x	A scalar. The amount of time after the censoring time cens.
tms	A numeric vector, with values sorted in ascending order. The event times to fit the RHawkes point process model.
cens	A scalar. The censoring time.
par	A numeric vector. Contains the parameters of the model, in order of the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.
h.fn	A (vectorized) function. The offspring density function.

`mu.fn` A (vectorized) function. The immigration hazard function.

`H.fn` A (vectorized) function. Its value at `t` gives the integral of the offspring density function from 0 to `t`.

`Mu.fn` A (vectorized) function. Its value at `t` gives the integral of the immigrant hazard function from 0 to `t`.

Value

The predictive density of the next event evaluated at `x`.

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
data(quake);
tms <- sort(quake$time);
# add some random noise to the identical event times
tms[213:214] <- tms[213:214] +
                    sort(c(runif(1, -1, 1)/(24*60), runif(1, -1, 1)/(24*60)))
curve(pred.den(x, tms = tms, cens = 35064, par= c(0.314, 22.2, 1266, 0.512))
        ,0 ,2000, col = 2, lty = 2)
```

pred.haz *RHawkes predictive hazard function*

Description

Calculates the predictive hazard function of the next observed event time after the censoring time `cens` based on observations over the interval [0,`cens`].

Usage

```
pred.haz(x, tms, cens, par,
        h.fn = function(x, p) dexp(x, rate = 1 / p),
        mu.fn = function(x, p) {
          exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
        pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
          },
        H.fn = function(x, p) pexp(x, rate = 1/p),
        Mu.fn = function(x, p) {
       - pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE)
        })
```

Arguments

x	A scalar. The amount of time after the censoring time cens.
tms	A numeric vector, with values sorted in ascending order. The event times to fit the RHawkes point process model.
cens	A scalar. The censoring time.
par	A numeric vector. Contains the parameters of the model, in order of the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.
h.fn	A (vectorized) function. The offspring density function.
mu.fn	A (vectorized) function. The immigration hazard function.
H.fn	A (vectorized) function. Its value at t gives the integral of the offspring density function from 0 to t.
Mu.fn	A (vectorized) function. Its value at t gives the integral of the immigrant hazard function from 0 to t.

Value

The predictive hazard rate of the next event evaluated at x.

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
data(quake);
tms <- sort(quake$time);
# add some random noise to the identical event times
tms[213:214] <- tms[213:214] +
                    sort(c(runif(1, -1, 1)/(24*60), runif(1, -1, 1)/(24*60)))
curve(pred.haz(x, tms = tms, cens = 35064, par= c(0.314, 22.2, 1266, 0.512))
        ,0 ,2000, col = 2, lty = 2)
```

quake *An RHawkes earthquake data set*

Description

An earthquake data set containing the earthquake occurrence times near the Japan region previously examined by Ogata (1998).

Usage

```
data(quake)
```

Format

The format is a vector of the arrival/birth times of earthquakes.

Details

Times of arrivals of earthquake occurrences in a vector in ascending order.

Source

Simulated by a call to the function `simHawkes1`.

Examples

```
## Not run:
data(quake)
## number of earthquake occurrences
nrow(quake)

## End(Not run)
```

---

sim.pred *Simulate a fitted RHawkes process model*

Description

Simulate a fitted RHawkes process model after the censoring time `cens` to a future time point `cens.tilde`.

Usage

```
sim.pred(tms, re.dist = rweibull, par,
    par.redist = list(shape = par[1], scale = par[2]),
    h.fn = function(x, p) dexp(x, rate = 1 / p), p.ofd = par[3],
    branching.ratio = par[4], cens, cens.tilde = cens * 1.5,
    mu.fn = function(x, p) {
        exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
       pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
    })
```

Arguments

`tms`	A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.
`re.dist`	A (vectorized) function. The immigrant renewal distribution function.
`par`	A numeric vector, giving the parameters of the model with the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.
`par.redist`	A numeric vector. The parameters of the immigrant renewal distribution.

`h.fn` A (vectorized) function. The offspring density function.

`p.ofd` A (named) list. The parameters of the offspring density.

`branching.ratio` A scalar. The branching ratio parameter.

`cens` A scalar. The censoring time.

`cens.tilde` A scalar. The future time that the simulation run until.

`mu.fn` A (vectorized) function. The immigration hazard function.

Value

A numeric vector that contains the simulated event times from censoring time `cens` up until `cens.tilde`

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
N <- 5; i <- 0;
data(quake); tms <- sort(quake$time);
# add some random noise the simultaneous occurring event times
tms[213:214] <- tms[213:214] +
                    sort(c(runif(1, -1, 1)/(24*60), runif(1, -1, 1)/(24*60)))
# simulate future event time based on MLE fitted RHawkes model
times <- replicate(N,
                   {cat(i<<-i+1,'\n');
                   sim.pred(tms = tms, par = c(0.314, 22.2, 1266, 0.512),
                            cens=35063)
                   })
plot(NA,NA,xlim=c(0,35063*1.5),ylim=c(0,max(lengths(times))+nrow(quake)),
     xlab="time",ylab="Sample path")
lines(c(0,quake$time),0:nrow(quake),type="s")
for(i in 1:N)
    lines(c(tail(quake$time,1),times[[i]]),nrow(quake)+0:length(times[[i]]),
          type="s",lty=2)
```

`sim.pred1` *Simulate a fitted RHawkes process model for prediction purposes*

Description

Simulate a fitted RHawkes process model from the censoring time `cens` to a future time point `cens.tilde`, conditional on the observed event times until the censoring time.

Usage

```
sim.pred1(tms, par, re.dist = rweibull,
    par.redist = list(shape = par[1], scale = par[2]),
    of.dis="exp", par.ofdis = list(rate=par[3]),
    branching.ratio = par[4], cens=tail(tms,1)+mean(diff(tms))/2,
    cens.tilde = cens * 1.5,
    mu.fn = function(x, p) {
        exp(dweibull(x, shape = p[1], scale = p[2], log = TRUE) -
      pweibull(x, shape = p[1], scale = p[2], lower.tail = FALSE, log.p = TRUE))
    })
```

Arguments

tms	A numeric vector, with values sorted in ascending order. Event times to fit the RHawkes point process model.
par	A numeric vector, giving the parameters of the model with the immigration parameters $\mu(.)$, offspring parameters $h(.)$ and lastly the branching ratio $\eta(.)$.
re.dist	A (vectorized) function. The function to simulate from the immigrant waiting times distribution.
par.redist	A (named) list, giving the parameters of the immigrant waiting time distribution.
of.dis	A character string, for the name of the offspring birth time distribution.
par.ofdis	A (named) list, giving the parameters of the offspring birth time distribution.
branching.ratio	A scalar in [0,1), the branching ratio parameter.
cens	A scalar. The censoring time.
cens.tilde	A scalar. The future time to run the simulation to.
mu.fn	A (vectorized) function. The immigration hazard function.

Value

A numeric vector that contains the simulated event times from censoring time `cens` up until `cens.tilde`

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

See Also

`sim.pred`.

Examples

```
N <- 5; i <- 0;
data(quake); tms <- sort(quake$time);
# add some random noise the simultaneous occurring event times
tms[213:214] <- tms[213:214] +
                  sort(c(runif(1, -1, 1)/(24*60), runif(1, -1, 1)/(24*60)))
```

```
# simulate future event time based on MLE fitted RHawkes model
times <- replicate(N,
                   {cat(i<<-i+1,'\n');
                   sim.pred1(tms = tms, par = c(0.314, 22.2, 1266, 0.512),
                             cens=35063)
                   })
plot(NA,NA,xlim=c(0,35063*1.5),ylim=c(0,max(lengths(times))+nrow(quake)),
     xlab="time",ylab="Sample path")
lines(c(0,quake$time),0:nrow(quake),type="s")
for(i in 1:N)
    lines(c(tail(quake$time,1),times[[i]]),nrow(quake)+0:length(times[[i]]),
          type="s",lty=2)
```

simRHawkes *Simulate a renewal Hawkes (RHawkes) process*

Description

Simulate a renewal Hawkes (RHawkes) process with given renewal immigration distribution function, offspring density function and branching ratio.

Usage

```
simRHawkes(re.dist = rweibull, par.redist = list(shape = 1, scale = 1),
           ofspr.den = function(x, p.ofd) 1 / p.ofd * exp(-x / p.ofd),
           p.ofd = 1, branching.ratio = 0.5, cens = 1, B = 10, B0 = 50,
           max.ofspr.den = max(optimize(ofspr.den, c(0, cens), maximum = TRUE,
           p = p.ofd)$obj, ofspr.den(0, p.ofd), ofspr.den(cens, p.ofd)) * 1.1)
```

Arguments

`re.dist`	A (vectorized) function. The immigrant renewal distribution function.
`par.redist`	A numeric vector. The parameters of the immigrant renewal distribution.
`ofspr.den`	A (vectorized) function. The offspring density function.
`p.ofd`	A numeric vector. The parameters of the offspring density.
`branching.ratio`	A scalar. The branching ratio parameter.
`cens`	A scalar. The censoring time.
`B`	A scalar. Tuning parameter for simulation of further immigrants.
`B0`	A scalar. Tuning parameter for simulation of initial immigrants.
`max.ofspr.den`	A scalar. The maximum value of the offspring density function from 0 to `cens`.

Details

The function works by simulating the arrival times of immigrants according to the renewal immigration distribution. The birth times of offspring from each immigrant are then simulated according to an inhomogeneous Poisson processes with appropriate intensity functions.

Value

A numeric vector of all pooled events (immigration/birth) times of all generations 0, 1, ...

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
B <- 10; i <- 0;
tms <- replicate(B,
                 {cat(i<<-i+1,'\n');
                 simRHawkes(par.redist = list(shape = 3, scale = 1),
                     p.ofd = 0.5, branching.ratio = 0.5,
                     cens = 100)
                 })
```

---

simRHawkes1 *Simulate a renewal Hawkes (RHawkes) process*

Description

Simulate a renewal Hawkes (RHawkes) process with given renewal immigration distribution function, offspring density function and branching ratio.

Usage

```
simRHawkes1(re.dist = rweibull, par.redist = list(shape = 1, scale = 1),
            of.dis = "exp", par.ofdis = list(rate=1),
            branching.ratio = 0.5, cens = 1, B = 10, B0 = 50,
            flatten=TRUE)
```

Arguments

`re.dist` A (vectorized) function. The immigrant renewal distribution function.

`par.redist` A numeric vector. The parameters of the immigrant renewal distribution.

`of.dis` A character string indicating the distribution for the offspring birth times, which has to be the 'distname' part of the `rdistname` functions for simulating postive random variables implemented in R, such as "exp", "weibull", "gamma", "lnorm", etc.

`par.ofdis` A list with named elements, giving the list of parameters of the offspring distribution, such as list(rate=1), list(shape=1,scale=1), etc.

`branching.ratio` A scalar between 0 and 1, the branching ratio parameter.

`cens` A scalar. The censoring time.

`B` A scalar. Tuning parameter for simulation of further immigrants.

`B0`	A scalar. Tuning parameter for simulation of initial immigrants.
`flatten`	A boolean scalar, which indicates whether the output events times should be flattened into an increasing sequence of times, or not (in which case the output is the immigrant arrival times, and the offspring birth times for different immigrants).

Details

The function works by simulating the arrival times of immigrants according to the renewal immigration distribution. The birth times of offspring from each immigrant are then simulated according to an inhomogeneous Poisson processes with appropriate intensity functions.

Value

A numeric vector of pooled event (immigration/offspring birth) times of all generations 0, 1, ..., if `flatten=TRUE`; A list with two components: `immitimes` for immigrant arrival times, and `offspringtimes` for birth times of offspring due to different immigrants.

Author(s)

Feng Chen <feng.chen@unsw.edu.au> Tom Stindl <t.stindl@unsw.edu.au>

Examples

```
tms <- simRHawkes1(par.redist = list(shape = 3, scale = 1),
                   par.ofdis = list(rate=0.5), branching.ratio = 0.5,
                   cens = 50)
plot(stepfun(tms,0:length(tms)),do.points=FALSE,vertical=FALSE,xlim=c(0,50))

tms.clust <- simRHawkes1(par.redist = list(shape = 3, scale = 1),
                         par.ofdis = list(rate=0.5), branching.ratio = 0.5,
                         cens = 50,flatten=FALSE)
plot(c(0,50),c(0, 1+(nt <-length(it <- tms.clust$immitimes))),
     type="n",xlab="time",ylab="cluster")
segments(x0 = it,y0=-0.2,y1=0.2)
for(i in 1:nt)
    segments(x0 = c(it[i],it[i]+tms.clust$offspringtimes[[i]]),
             y0=i-0.2,y1=i+0.2)
abline(h=0:(nt+1),col="light gray",lty=2)
segments(x0=unlist(lapply(1:nt,function(i)c(it[i],it[i]+tms.clust$offspringtimes[[i]]))),
         y0=nt+1-0.2,y1=nt+1+0.2)
```

`tms` *mid-price change times of the AUD/USD exchange rate*

Description

A financial data set containing the mid-price changes of the AUD/USD foreign exchange rate during the trading week from 20:00:00 GMT on Sunday 19/07/2015 to 21:00:00 GMT Friday 24/07/2015.

Usage

```
data(tms)
```

Format

The format is a list of the arrival times of mid price changes that occur every hour in 121 non-overlapping windows.

Details

Times of arrivals of mid-price changes is listed together in ascending order.

Source

Simulated by a call to the function `simHawkes1`.

Examples

```
data(tms)
## number of non over-lapping hourly windows
length(tms)
```

Index