

Package ‘RHawkes’

November 17, 2017

Type Package

Title Renewal Hawkes Process

Version 0.0

Date 2017-11-16

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

Thomas Stindl <t.stindl@unsw.edu.au>

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

Description Simulate a renewal Hawkes (RHawkes) self-exciting process, with a given immigrant hazard rate function and offspring density function. Calculate the likelihood of a RHawkes process with given hazard rate function and offspring density function for an (increasing) sequence of event times. Calculate the Rosenblatt residuals of the event times. Predict future event times based on observed event times up to a given time. For details see Chen and Stindl (2017) <doi:10.1080/10618600.2017.1341324>.

License GPL (>= 2)

Depends R (>= 2.10), IHSEP

NeedsCompilation no

Repository CRAN

Date/Publication 2017-11-17 11:48:19 UTC

R topics documented:

RHawkes-package	2
EM1partial	3
EM2partial	4
mllRH	6
mllRH1	8
mllRH2	9
pred.den	11
pred.haz	12
quake	13
sim.pred	14

simRHawkes	15
tms	17

Index	18
--------------	-----------

RHawkes-package	<i>Renewal Hawkes Process</i>
-----------------	-------------------------------

Description

Simulate a renewal Hawkes (RHawkes) self-exciting process, with a given immigrant hazard rate function and offspring density function. Calculate the likelihood of a RHawkes process with given hazard rate function and offspring density function for an (increasing) sequence of event times. Calculate the Rosenblatt residuals of the event times. Predict future event times based on observed event times up to a given time. For details see Chen and Stindl (2017) <doi:10.1080/10618600.2017.1341324>.

Details

The DESCRIPTION file:

```
Package:      RHawkes
Type:        Package
Title:       Renewal Hawkes Process
Version:     0.0
Date:       2017-11-16
Author:      Feng Chen <feng.chen@unsw.edu.au> Thomas Stindl <t.stindl@unsw.edu.au>
Maintainer:  Feng Chen <feng.chen@unsw.edu.au>
Description: Simulate a renewal Hawkes (RHawkes) self-exciting process, with a given immigrant hazard rate function and
License:     GPL (>=2)
Depends:     IHSEP
```

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
m1LRH	Minus loglikelihood of a RHawkes model
m1LRH1	Minus loglikelihood of a RHawkes model with parent probabilities
m1LRH2	Minus loglikelihood of a RHawkes model with Rosenblatt residuals
pred.den	RHawkes predictive density function
pred.haz	RHawkes predictive hazard function
quake	Japanese Earthquake Data
sim.pred	Simulate a fitted RHawkes process model

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

```
## simulate data
set.seed(1)
tms <- simRHawkes(cens=10)
## the slower version of the EM algorithms on simulated data with
## default immigrant hazard function and offspring density
system.time( est1 <- EM1partial(tms=tms, cens=50, pars=c(1,1,0.5,1)) )
```

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 $\mu(\cdot)$, offspring parameters $h(\cdot)$ and lastly the branching ratio $\eta(\cdot)$.
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

```

## simulate data
set.seed(1)
tms <- simRHawkes(cens=10)
## the slower version of the EM algorithms on simulated data with
## default immigrant hazard function and offspring density
system.time( est2 <- EM2partial(tms=tms, cens=50, pars=c(1,1,0.5,1)) )

```

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 $\mu(\cdot)$, offspring parameters $h(\cdot)$ and lastly the branching ratio $\eta(\cdot)$.
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

```

## simulate data from a RHawkes process (Hawkes, by the defaulting
## parameters)
set.seed(1)
ce <- 50;
tms <- simRHawkes(cens=ce,par.redist=list(shape=1,scale=1.5),branching.ratio=0.3)
## calculate the MLE when data is treated as Hawkes process event
## times on [0,ce]
sim.est0 <- optim(c(1.5,1,0.3),
  function(p){
    mloglik1d(jtms=tms,TT=ce,
      nu=function(x){rep(1/p[1],length(x))},
      gcoef=c(p[3]*p[2],p[2]),
      Inu=function(x){x/p[1]})
  },
  control=list(maxit=5000,trace=TRUE,reltol=1e-9),
  )
## calculate the MLE when data is treated as RHawkes process event
## times on [0,ce]
sim.est1 <- optim(c(1,sim.est0$par),
  mllRH,tms=tms,cens=ce,
  control=list(maxit = 5000, trace = TRUE,reltol=1e-9),
  hessian=TRUE)

sim.est1$par

## 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 respectivley
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

## Not run:
set.seed(1)
ce <- 2000;

```

```

tms <- simRHawkes(cens=ce,par.redist=list(shape=1,scale=2),branching.ratio=0.4)
## calculate the MLE when data is treated as Hawkes process event
## times on [0,ce]
sim.est0 <- optim(c(2,1,0.4),
  function(p){
    mloglik1d(jtms=tms,TT=ce,
      nu=function(x){rep(1/p[1],length(x))},
      gcoef=c(p[3]*p[2],p[2]),
      Inu=function(x){x/p[1]})
  },
  control=list(maxit=5000,trace=TRUE,reltol=1e-9),
  )
## calculate the MLE when data is treated as Weibull RHawkes
## process event times on [0,ce]
system.time(sim.est1 <- optim(c(1,sim.est0$par),
  m1LRH,tms=tms,cens=ce,
  control=list(maxit = 5000, trace = TRUE,reltol=1e-9),
  hessian=TRUE))

sim.est1$par

## End(Not run)

```

m1LRH1

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 m1LRH although this version also returns the parent probabilities.

Usage

```

m1LRH1(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 $\mu(\cdot)$, offspring parameters $h(\cdot)$ and lastly the branching ratio $\eta(\cdot)$.
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 (on the log scale)
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(10,0,100)), 100, 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

```
mlRH2(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 $\mu(\cdot)$, offspring parameters $h(\cdot)$ and lastly the branching ratio $\eta(\cdot)$.
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

mlRH

Examples

```

tmp <- mllRH2(sort(runif(50,0,100)),100,c(1,2,2,0.05))
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",...)
}
par(mfrow=c(1,2))
qqunif(tmp$U)
acf(tmp$U)
ks.test(tmp$U,"punif")

```

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 $[0, \text{cens}]$.

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

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

Japanese Earthquake Data

Description

Times and magnitudes (Richter scale) of 483 shallow earthquakes in an offshore region east of Honshu and south of Hokkaido, for the period 1885-1980.

Usage

```
data("quake")
```

Format

A data frame with 483 observations on the following 2 variables.

time a numeric vector giving earthquake times in days since start of 1885

mag a numeric vector giving the magnitudes of the earthquakes

Details

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

Source

Simulated by a call to the function `simHawkes1`.

Examples

```
data(quake)
summary(quake)
```

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

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

h.fn	A (vectorized) function. The offspring density function.
p.ofd	A numeric vector. The paramters 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$time);
# add some random noise the simultaneous occuring event times
tms[213:214] <- tms[213:214] +
  sort(c(runif(1, -1, 1)/(24*60), runif(1, -1, 1)/(24*60)))
# simulate future event time based on MLE fitted RHawkes model
times <- replicate(N,
  {cat(i<<-i+1, '\n');
  sim.pred(tms = tms, par = c(0.314, 22.2, 1266, 0.512),
    cens=35063)
  })
plot(NA, NA, xlim=c(0, 35063*1.5), ylim=c(0, max(lengths(times))+nrow(quake)),
  xlab="time", ylab="Sample path")
lines(c(0, quake$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)

```

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

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

tms	<i>mid-price change times of the AUD/USD exchange rate</i>
-----	--

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

- *Topic **EM algorithm**
 - EM1partial, 3
 - EM2partial, 4
 - *Topic **RHawkes**
 - EM1partial, 3
 - EM2partial, 4
 - m11RH, 6
 - *Topic **Rosenblatt residual**
 - m11RH2, 9
 - *Topic **datasets**
 - quake, 13
 - tms, 17
 - *Topic **goodness-of-fit**
 - m11RH2, 9
 - *Topic **likelihood**
 - m11RH, 6
 - m11RH1, 8
 - *Topic **package**
 - RHawkes-package, 2
 - *Topic **parent probabilities**
 - m11RH1, 8
 - *Topic **point process**
 - m11RH2, 9
 - pred.den, 11
 - pred.haz, 12
 - *Topic **prediction**
 - pred.den, 11
 - pred.haz, 12
 - *Topic **predict**
 - sim.pred, 14
 - *Topic **renewal process**
 - simRHawkes, 15
 - *Topic **self-exciting**
 - simRHawkes, 15
 - *Topic **simulations**
 - sim.pred, 14
- m11RH, 6
m11RH1, 8
m11RH2, 9
- pred.den, 11
pred.haz, 12
- quake, 13
- RHawkes (RHawkes-package), 2
RHawkes-package, 2
- sim.pred, 14
simRHawkes, 15
- tms, 17
- EM1partial, 3
EM2partial, 4