# Package 'hawkesbow'

October 13, 2022

**Title** Estimation of Hawkes Processes from Binned Observations

**Version** 1.0.2

**Description** Implements an estimation method for Hawkes processes when count data are only observed in discrete time, using a spectral approach derived from the Bartlett spectrum, see Cheysson and Lang (2020) <arXiv:2003.04314>. Some general use functions for Hawkes processes are also included: simulation of (in)homogeneous Hawkes process, maximum likelihood estimation, residual analysis, etc.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.0

**Depends** Rcpp

**Imports** methods, nloptr

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

**LinkingTo** Rcpp, RcppArmadillo, BH

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Felix Cheysson [aut, cre] (<https://orcid.org/0000-0002-9095-2878>)

**Maintainer** Felix Cheysson <felix@cheysson.fr>

**Repository** CRAN

**Date/Publication** 2021-04-09 23:40:05 UTC

## R topics documented:

---

`compensator` *Compensator of a Hawkes process*

---

## Description

Outputs the compensator (integrated intensity) of a Hawkes process.

## Usage

```
compensator(x, t, fun = NULL, repr = NULL, family = NULL, M = NULL, ...)
```

## Arguments

| | |
|---|---|
| `x` | A non-negative numeric vector, sorted in ascending order; or an object of class "hawkes" output by function `hawkes`. |
| `t` | A non-negative numeric value or vector, at which the intensity should be computed. |
| `fun` | (default = TRUE) A non-negative numeric function or value - intensity (function) of the immigrant process. |
| `repr` | (default = NULL) A non-negative numeric value - mean number of offsprings. |
| `family` | (default = NULL) A character string "name" naming a distribution with corresponding distribution function `dname`, or directly the distribution function. |
| `M` | (default = NULL) A non-negative numeric value - upper bound on `fun`(ignored if `fun` is a numeric value). |
| `...` | Additional arguments passed on to the random generation function `dname`. |

## Value

The compensator at time t.

## Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1,
# reproduction mean 0.5 and exponential fertility distribution with rate 2.
x <- hawkes(10, fun=1, repr=0.5, family="exp", rate=2)
compensator(x, 0:10)
# Compensator with a different set of parameters
compensator(x, 0:10, repr=0.8, rate=3)
# Compensator with a different distribution function
compensator(x, 0:10, family="chisq", df=2)
# Simulate a Hawkes process with baseline intensity function 1 + sin(x),
# reproduction mean 0.5 and custom [0,1]-triangular fertility function.
x <- hawkes(10, fun=function(y) {1+sin(y)}, M=2, repr=0.5,
            family=function(n) {1 - sqrt(1 - runif(n))})
compensator(x, 0:10, family=function(y) ifelse(y>0 & y<1, 2-2*y, 0))
```

---

# discrete *Discretizes a Hawkes simulation*

---

## Description

Discretizes a Hawkes simulation

## Usage

```
discrete(hawkes, length = NULL, binsize = NULL)
```

## Arguments

| | |
|---|---|
| hawkes | An object created by the function hawkes |
| length | (Either) The length for the output vector |
| binsize | (Either) The binsize for the discretization |

## Value

The vector of counts

## Examples

```
x = hawkes(100, fun=1, repr=0.5, family="exp", rate=2)
y = discrete(x, length=100)
z = discrete(x, binsize=1)
all(y == z)
```

dpowerlaw *The power law distribution*

## Description

Density, distribution function, quantile function and random generation for the power law distribution with shape equal to shape and scale equal to scale.

## Usage

```
dpowerlaw(x, shape = 1, scale = 1)

ppowerlaw(q, shape = 1, scale = 1)

qpowerlaw(p, shape = 1, scale = 1)

rpowerlaw(n, shape = 1, scale = 1)
```

## Arguments

| | |
|---|---|
| x, q | vector of quantiles. |
| shape | parameter of shape. |
| scale | parameter of scale. |
| p | vector of probabilities. |
| n | number of observations. |

## Details

The density function of the power law distribution is

$$f(t) = \theta a^{\theta}(a+t)^{-\theta-1}$$

where $\theta$ is the shape parameter, and $a$ the scale parameter.

## Value

dpowerlaw gives the density, ppowerlaw gives the distribution function, qpowerlaw gives the quantile function, and rpowerlaw generates random deviates.

---

`E1_imaginary` *Exponential integral of imaginary argument*

---

## Description

Calculates the value of

$$E_1(ix) = \int_1^\infty \frac{e^{-ixt}}{t} \mathrm{d}t$$

using its relation to the trigonometric integrals (cf. https://en.wikipedia.org/wiki/Exponential_integral#Exponential_integral_of_imaginary_argument):

$$E_1(ix) = i\left[-\frac{1}{2}\pi + Si(x)\right] - Ci(x)$$

and their Pad\'e approximants (cf. https://en.wikipedia.org/wiki/Trigonometric_integral#Efficient_evaluation)

## Usage

```
E1_imaginary(x)
```

## Arguments

| | |
|---|---|
| `x` | A non-negative number |

## Value

The exponential integral of argument ix

## Examples

```
E1_imaginary(1.0)
```

---

`Etheta_imaginary` *Incomplete gamma function of imaginary argument with arbitrary power*

---

## Description

Calculates the value of

$$-ixe^{ix}E_\theta(ix) = -ixeix\int_1^\infty t^{-\theta}e^{-ixt}\mathrm{d}t$$

for $\theta > 0$. This is achieved using recursive integrations by parts until $0 < \theta \leq 1$, then using either the exponential integral `E1_imaginary` if $\theta = 1$, or the incomplete gamma function `inc_gamma_imag` if $0 < \theta < 1$.

### Usage

```
Etheta_imaginary(theta, x)
```

### Arguments

| | |
|---|---|
| theta | A strictly positive number |
| x | A vector of non-negative numbers |

### Value

The incomplete gamma function of imaginary argument with arbitrary power (see Details)

### Examples

```
Etheta_imaginary(3.14, 1.0)
```

---

## Exponential *Reproduction kernels for the Hawkes processes*

---

### Description

These classes are derived from the class Model, each implementing a different reproduction kernel for the Hawkes process. They inherit all fields from Model.

### Details

- The kernel Exponential has density function

$$h^*(t) = \beta \exp(-\beta t) 1_{\{t \geq 0\}}.$$

  Its vector of parameters must be of the form $(\eta, \mu, \beta)$. Both loglik, its derivatives, and whittle can be used with this reproduction kernel.

- The kernel SymmetricExponential has density function

$$h^*(t) = 0.5\beta \exp(-\beta |t|).$$

  Its vector of parameters must be of the form $(\eta, \mu, \beta)$. Only whittle can be used with this reproduction kernel.

- The kernel Gaussian has density function

$$h^*(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t-\nu)^2}{2\sigma^2}\right).$$

  Its vector of parameters must be of the form $(\eta, \mu, \nu, \sigma^2)$. Only whittle is available with this reproduction kernel.

- The kernel `PowerLaw` has density function

$$h^*(t) = \theta a^\theta (t+a)^{-\theta-1} 1_{\{\theta>0\}}.$$

  Its vector of parameters must be of the form $(\eta, \mu, \theta, a)$. Both `loglik`, its derivatives, and `whittle` can be used with this reproduction kernel.

- The kernels `Pareto3`, `Pareto2` and `Pareto1` have density function

$$h^*_\theta(t) = \theta a^\theta t^{-\theta-1} 1_{\{t>a\}},$$

  with $\theta$ = 3, 2 and 1 respectively. Their vectors of parameters must be of the form $(\eta, \mu, a)$. Only `whittle` is available with this reproduction kernel.

## See Also

Model

---

`hawkes` *Simulation of a Hawkes process*

---

## Description

Simulates a Hawkes process using its cluster representation:

- First, the immigrants are drawn according to an (inhomogeneous) Poisson process with intensity measure `fun`.
- Second, the number of offsprings of an immigrant follows a Poisson distribution with intensity `repr`.
- Third, these offsprings are distributed according to the `family` distribution.
- Then, generate further offsprings according to the last two steps.

## Usage

```
hawkes(end, fun, repr, family, M = NULL, ...)
```

## Arguments

| | |
|---|---|
| `end` | A non-negative numeric value - right bound of the interval $[0, \mathrm{end}]$. |
| `fun` | A non-negative function or numeric value - intensity (function) of the immigrant process. |
| `repr` | A non-negative numeric value - mean number of offsprings. |
| `family` | A character string "name" naming a distribution with corresponding random generation function `rname`, or directly the random generation function. |
| `M` | (default = NULL) A non-negative numeric value - upper bound on `fun`(ignored if `fun` is a numeric value). |
| `...` | Additional arguments passed on to the random generation function. |

## Value

A S3 object of class Hawkes containing a vector ($p) of simulated values, and all other objects used for the simulation.

## Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1,
# reproduction mean 0.5 and exponential fertility function with rate 2.
x <- hawkes(10, fun=1, repr=0.5, family="exp", rate=2)
# Simulate a Hawkes process with baseline intensity function 1 + sin(x),
# reproduction mean 0.5 and custom [0,1]-triangular fertility function.
x <- hawkes(10, fun=function(y) {1+sin(y)}, M=2, repr=0.5,
            family=function(n) {1 - sqrt(1 - runif(n))})
```

---

# hawkes_ogata *Simulation of a Hawkes process*

---

## Description

Simulates a Hawkes process via Ogata's modified thinning algorithm on $[0, \mathrm{end}]$. This is less efficient than function hawkes, but can be useful for pedagogical reasons.

## Usage

```
hawkes_ogata(end, lambda, alpha, beta, lambda0 = NULL)
```

## Arguments

| | |
|---|---|
| end | Right bound on time. |
| lambda | Baseline intensity. |
| alpha | Parameter for the amplitude of the spike. |
| beta | Parameter for the speed of exponential decay. |
| lambda0 | (Optional) Initial value of the conditional intensity. |

## Value

A S3 object of class Hawkes containing a vector ($p) of simulated values, and all other objects used for the simulation.

## Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1 and
# excitation function 1*exp(-2t)
x <- hawkes_ogata(10, 1, 1, 2)
plot(x)
```

`inc_gamma_imag` *Incomplete gamma function of imaginary argument*

## Description

Calculates the value of

$$\Gamma_1(x,\alpha) = \int_x^\infty t^{\alpha-1}e^{-it}\mathrm{d}t$$

for $0 < \alpha < 1$ through the following relations:

$$\int_0^\infty t^{\alpha-1}e^{-it}\mathrm{d}t = e^{-i\frac{\pi}{2}\alpha}\int_0^\infty t^{\alpha-1}e^{-t}\mathrm{d}t = e^{-i\frac{\pi}{2}\alpha}\Gamma(\alpha).$$

obtained by contour integration, and:

$$\int_0^x t^{\alpha-1}e^{-it}\mathrm{d}t = \int_0^x t^{\alpha-1}\cos(t)\mathrm{d}t - i\int_0^x t^{\alpha-1}\sin(t)\mathrm{d}t = Ci(x,\alpha) - iSi(x,\alpha)$$

. The first integral is calculated using function "tgamma" from the library "boost::math", while the functions Ci and Si are approximated via Taylor expansions.

## Usage

```
inc_gamma_imag(x, alpha)
```

## Arguments

| | |
|---|---|
| `x` | A non-negative number |
| `alpha` | A number between 0 and 1 (strictly) |

## Value

The incomplete gamma function of imaginary argument (see Details)

## Examples

```
inc_gamma_imag(1.0, 0.5)
```

## inhpois *Simulation of an inhomogeneous Poisson process by thinning*

### Description

Simulates an inhomogeneous Poisson process via Ogata's modified thinning algorithm on $[0, \mathrm{end}]$. An homogeneous Poisson process with intensity `M` is first generated on $[0, \mathrm{end}]$, then thinned using the specified intensity function `fun`.

### Usage

```
inhpois(end, fun, M = NULL)
```

### Arguments

| | |
|---|---|
| `end` | A non-negative numeric value - right bound of the interval $[0, \mathrm{end}]$. |
| `fun` | A non-negative function or numeric value - intensity (function) of the Poisson process. |
| `M` | (default = NULL) A non-negative numeric value - upper bound on `fun` (ignored if `fun` is a numeric value). |

### Value

A S3 object of class `inhpois` containing a vector ($p) of simulated values, and all other objects used for the simulation.

### Examples

```
# Simulate an inhomogeneous Poisson process with function intensity 1 + sin(x) (bounded by 2)
x <- inhpois(end=10, fun=function(y) {1 + sin(y)}, M=2)
# Simulate a homogeneous Poisson process with intensity 3
x <- inhpois(end=10, fun=3)
```

## intensity *Intensity of a Hawkes process*

### Description

Outputs the intensity of a Hawkes process `x`, given the specified set of parameters.

### Usage

```
intensity(x, t, fun = NULL, repr = NULL, family = NULL, M = NULL, ...)
```

## Arguments

| | |
|---|---|
| x | A non-negative numeric vector, sorted in ascending order; or an object of class "hawkes" output by function hawkes. |
| t | A non-negative numeric value or vector, at which the intensity should be computed. |
| fun | (default = TRUE) A non-negative numeric function or value - intensity (function) of the immigrant process. |
| repr | (default = NULL) A non-negative numeric value - mean number of offsprings. |
| family | (default = NULL) A character string "name" naming a distribution with corresponding distribution function dname, or directly the distribution function. |
| M | (default = NULL) A non-negative numeric value - upper bound on fun(ignored if fun is a numeric value). |
| ... | Additional arguments passed on to the random generation function dname. |

## Details

If the input x has been simulated using the function hawkes, the parameters of the simulation will be used by default to compute the intensity. If any parameter is specified in this function call, the function will use this instead.

## Value

The intensity at time t.

## Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1,
# reproduction mean 0.5 and exponential fertility distribution with rate 2.
x <- hawkes(10, fun=1, repr=0.5, family="exp", rate=2)
intensity(x, 0:10)
# Intensity with a different set of parameters
intensity(x, 0:10, repr=0.8, rate=3)
# Intensity with a different distribution function
intensity(x, 0:10, family="chisq", df=2)
# Simulate a Hawkes process with baseline intensity function 1 + sin(x),
# reproduction mean 0.5 and custom [0,1]-triangular fertility function.
x <- hawkes(10, fun=function(y) {1+sin(y)}, M=2, repr=0.5,
            family=function(n) {1 - sqrt(1 - runif(n))})
intensity(x, 0:10, family=function(y) ifelse(y>0 & y<1, 2-2*y, 0))
```

`mle` *Fitting Hawkes processes from continuous data*

## Description

This function fits a Hawkes process to continuous data by minimizing the likelihood on the interval $[0, \text{end}]$.

## Usage

```
mle(events, kern, end, init = NULL, opts = NULL, ...)
```

## Arguments

| | |
|---|---|
| `events` | The locations of events (sorted in ascending order) |
| `kern` | Either a string (partially) matching one of the kernels implemented (see Details), or an object of class Model |
| `end` | The time until which the process is observed. |
| `init` | (Optional) Initial values of the optimisation algorithm |
| `opts` | (Optional) To be passed to `nloptr` |
| `...` | Additional arguments passed to `nloptr` |

## Details

The maximum likelihood estimation procedure has only been implemented for the exponential and the power law kernels. For the exponential kernel, the likelihood is computed in $O(n)$ complexity (as described in details in T. Ozaki and Y. Ogata, "Maximum likelihood estimation of Hawkes' self-exciting point processes," Ann. Inst. Stat. Math., vol. 31, no. 1, pp. 145–155, Dec. 1979). For the power law kernel, the complexity is $O(n^2)$.

## Value

Returns a list containing the solution of the optimisation procedure, the object Model with its parameters updated to the solution, and the output produced by `nloptr`.

## See Also

`hawkes()` for the simulation of Hawkes processes, Model for the abstract class, and Exponential for the specific reproduction kernels.

## Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1,
# reproduction mean 0.5 and exponential fertility function with rate 2.
x = hawkes(100, fun = 1, repr = .5, family = "exp", rate = 1)
# Estimate the parameters from the arrival times of `x` using MLE
opt = mle(x$p, "Exponential", x$end)
opt$par                          # Estimated parameters
opt$model$ddloglik(x$p, x$end)     # Hessian matrix of the log-likelihood
```

---

`Model` *C++ abstract class for Hawkes processes*

---

## Description

This is a C++ abstract class for Hawkes processes, which holds methods for the estimation of its parameters.

## Details

This serves as a basis for the Hawkes model and its count sequence, with conditional intensity function

$$\lambda(t) = \eta + \mu \sum_{T_i < t} h^*(t - T_i).$$

As an abstract class, an object of class `Model` should never be directly instanciated, but rather one of its derived class. The constructor can take no argument, in which case the vector `param` is initialised to sensible values and `binsize` defaults to 1. Alternatively, `param` and/or `binsize` can be specified.

## Fields

`param` Vector of parameters of the Hawkes process, of the form $(\eta, \mu, ...)$.

`binsize` Bin size for the count sequences.

`new(DerivedClass,(param),(binsize))` Constructor for derived classes; `param` and/or `binsize` can be safely ignored.

`mean()` Returns the expected value on $[0, \mathrm{end}]$.

`dmean()` Returns the Jacobian matrix of the expected value on $[0, \mathrm{end}]$.

`ddmean()` Returns the Hessian matrix of the expected value on $[0, \mathrm{end}]$.

`f(xi)` Returns the spectral density function of the time-continuous count sequence.

- `xi` A numeric vector of frequencies.

`f1(xi,trunc)` Returns the spectral density function of the discrete time count sequence.

- `xi` A numeric vector of frequencies.
- `trunc` The number of foldings to take into account for the aliasing.

`whittle(I,trunc)` Returns the log-spectral likelihood of a discrete time count sequence.

- `I` The periodogram of the count sequence.

- `trunc` The number of foldings to take into account for the aliasing.

`loglik(events,end)` Returns the log-likelihood of a sequence of arrival times.

- `events` The sequence of arrival times.
- `end` The endpoint of the observation window $[0, \mathrm{end}]$.

`dloglik(events,end)` Returns the Jacobian matrix of the log-likelihood of a sequence of arrival times.

- `events` The sequence of arrival times.
- `end` The endpoint of the observation window $[0, \mathrm{end}]$.

`ddloglik(events,end)` Returns the Hessian matrix of the log-likelihood of a sequence of arrival times.

- `events` The sequence of arrival times.
- `end` The endpoint of the observation window $[0, \mathrm{end}]$.

## See Also

Exponential

## Examples

```
# Simulate 1000 exponential Hawkes processes on \eqn{[0, 100]},
# and average the periodogram of the count sequences with bin size 1
# at each frequency.
I = rep(0, 100)
for (k in 1:1e3) {
    x = hawkes(100, fun = 1, repr = .5, family = "exp", rate = 2)
    y = discrete(x, binsize = 1)
    I = I + Mod(fft(y - mean(y)))^2 / length(y)
}

# Check that the averaged periodogram correctly approximates the spectral
# density function of the count sequence
model = new(Exponential)
model$param = c(1, .5, 2)
model$binsize = 1

z = 2 * pi * 0:99 / 100     # Frequencies of the periodogram
plot(z, I / 1e3, type = "l") # Averaged periodogram
lines(z, model$f1(xi = z, trunc = 10L), col = "red")
```

---

`plot.hawkes` *Plot of a Hawkes process*

---

## Description

Plots the realisation of a Hawkes process and either its cluster representation (`intensity=FALSE`, only available for a simulated Hawkes process) or its intensity function (`intensity=TRUE`).

## Usage

```
## S3 method for class 'hawkes'
plot(
  x,
  intensity = FALSE,
  precision = 1000,
  fun = NULL,
  repr = NULL,
  family = NULL,
  M = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| x | Either: a numeric vector, sorted in ascending order; or an object of class "hawkes" output by function hawkes. |
| intensity | (default = FALSE) A boolean - whether to represent the cluster representation (FALSE) or the intensity function (TRUE). |
| precision | (default = 1e3) Number of points to plot. |
| fun | (default = NULL) A numeric function - intensity (function) of the immigrant process. |
| repr | (default = NULL) A non-negative numeric value - mean number of offsprings. |
| family | (default = NULL) A character string "name" naming a distribution with corresponding distribution function dname, or directly the distribution function. |
| M | (default = NULL) A non-negative numeric value - upper bound on fun(ignored if fun is a numeric value). |
| ... | Additional arguments passed on to the random generation function dname. |

## Value

None

## Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1,
# reproduction mean 0.5 and exponential fertility function with rate 2.
x <- hawkes(10, fun=1, repr=0.5, family="exp", rate=2)
plot(x)
# Simulate a Hawkes process with baseline intensity function 1 + sin(x),
# reproduction mean 0.5 and custom [0,1]-triangular fertility function.
x <- hawkes(10, fun=function(y) {1+sin(y)}, M=2, repr=0.5,
            family=function(n) {1 - sqrt(1 - runif(n))})
plot(x, intensity=TRUE, family=function(y) ifelse(y>0 & y<1, 2-2*y, 0))
```

## plot.hawkes_ogata *Plot of a simulated Hawkes process*

### Description

Plots a Hawkes process simulated by the function `hawkes_ogata`, highlighting the steps used in Ogata's thinning algorithm.

### Usage

```
## S3 method for class 'hawkes_ogata'
plot(x, precision = 1000, ...)
```

### Arguments

| | |
|---|---|
| `x` | A simulated Hawkes process from `hawkes_ogata`. |
| `precision` | (default = 1e3) Number of points to plot. |
| `...` | Only there to fit the declaration of S3 method `plot`. |

### Value

None

### Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1 and
# excitation function 1*exp(-2t)
x <- hawkes_ogata(10, 1, 1, 2)
plot(x)
```

## plot.inhpois *Plot of a simulated inhomogeneous Poisson process*

### Description

Plots a simulated inhomogeneous Poisson process, highlighting the steps used in Ogata's thinning algorithm.

### Usage

```
## S3 method for class 'inhpois'
plot(x, precision = 1000, ...)
```

### Arguments

| | |
|---|---|
| x | A simulated inhomogeneous Poisson process. |
| precision | (default = 1e3) Number of points to plot. |
| ... | Only there to fit the declaration of S3 method plot. |

### Value

None

### Examples

```
# Simulate an inhomogeneous Poisson process with function intensity 1 + sin(x)
x <- inhpois(end=10, fun=function(y) {1 + sin(y)}, M=2)
plot(x)
```

---

## residuals *Residuals of a Hawkes process*

---

### Description

Outputs the residuals (values of the compensator at the times of arrival) of a Hawkes process. Useful function for diagnosis through the random time change theorem: the residuals should follow a unit rate Poisson process.

### Usage

```
residuals(x, fun = NULL, repr = NULL, family = NULL, M = NULL, ...)
```

### Arguments

| | |
|---|---|
| x | A non-negative numeric vector, sorted in ascending order; or an object of class "hawkes" output by function hawkes. |
| fun | (default = TRUE) A non-negative numeric function or value - intensity (function) of the immigrant process. |
| repr | (default = NULL) A non-negative numeric value - mean number of offsprings. |
| family | (default = NULL) A character string "name" naming a distribution with corresponding distribution function dname, or directly the distribution function. |
| M | (default = NULL) A non-negative numeric value - upper bound on fun(ignored if fun is a numeric value). |
| ... | Additional arguments passed on to the random generation function dname. |

### Value

The residuals of the process.

## Examples

```
# Simulate an exponential Hawkes process with baseline intensity 1,
# reproduction mean 0.5 and exponential fertility distribution with rate 2.
x <- hawkes(10, fun=1, repr=0.5, family="exp", rate=2)
resid = residuals(x)
resid
plot(resid)
abline(0, 1, col="red", lty="dashed")
# Residuals with a different set of parameters
residuals(x, repr=0.8, rate=3)
# Residuals with a different distribution function
residuals(x, family="chisq", df=2)
# Simulate a Hawkes process with baseline intensity function 1 + sin(x),
# reproduction mean 0.5 and custom [0,1]-triangular fertility function.
x <- hawkes(10, fun=function(y) {1+sin(y)}, M=2, repr=0.5,
            family=function(n) {1 - sqrt(1 - runif(n))})
resid = residuals(x, family=function(y) ifelse(y>0 & y<1, 2-2*y, 0))
plot(resid)
abline(0, 1, col="red", lty="dashed")
```

---

`whittle` *Fitting Hawkes processes from discrete data*

---

## Description

This function fits a Hawkes process to discrete data by minimizing the Whittle contrast.

## Usage

```
whittle(counts, kern, binsize = NULL, trunc = 5L, init = NULL, ...)
```

## Arguments

| | |
|---|---|
| `counts` | A bin-count sequence |
| `kern` | Either a string (partially) matching one of the kernels implemented (see Details), or an object of class Model |
| `binsize` | (Optional) The bin size of the bin-count sequence; if omitted, defaults to 1 if `kern` is a string, or uses the member `binsize` of `kern` if it is of class Model |
| `trunc` | (Optional) The number of foldings taken into account due to aliasing |
| `init` | (Optional) Initial values of the optimisation algorithm |
| `...` | Additional arguments passed to `optim` |

## Details

If specified as string, the argument `kern` must match (partially) one of the following (upper cases not taken into account): Exponential, SymmetricExponential, Gaussian, PowerLaw, Pareto3, Pareto2, Pareto1. The periodogram used in the optimisation procedure is computed in complexity $O(n \log n)$, using function `fft`.

## Value

Returns a list containing the solution of the optimisation procedure, the object Model with its parameters updated to the solution, and the output produced by `optim`.

## See Also

`hawkes()` for the simulation of Hawkes processes, `discrete()` for the discretisation of simulated Hawkes processes, Model for the abstract class, and Exponential for the specific reproduction kernels.

## Examples

```
# Simulate and fit a Hawkes process with exponential kernel
x = hawkes(1000, fun = 1, repr = .5, family = "exp", rate = 1)
y = discrete(x, binsize = 1)
opt = whittle(y, "Exponential")
opt$par      # Estimated parameters


# May take up to 20 seconds
# Simulate and fit a Hawkes process with power law kernel
x = hawkes(1000, fun = 1, repr= .3, family = "powerlaw", shape = 3.5, scale = 1.0)
y = discrete(x, binsize = 1)
opt = whittle(y, "powerlaw")
opt$par      # Estimated parameters
```

# Index