

# Package ‘CTRE’

May 7, 2018

**Type** Package

**Title** Thresholding Bursty Time Series

**Version** 0.1.0

**Date** 2018-04-25

**Maintainer** Peter Straka <p.straka@unsw.edu.au>

**Description** Models extremes of 'bursty' time series via  
Continuous Time Random Exceedances (CTRE).  
See <arXiv:1802.05218>, K. Hees, S. Nayak, P.Straka, 2018.

**URL** <https://unsw-math.github.io/CTRE/>

**BugReports** <https://github.com/UNSW-MATH/CTRE/issues>

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Imports** assertthat, graphics, MittagLeffleR, plyr, stats, tea, utils,  
zoo

**Depends** R (>= 2.10)

**Suggests** evmix, magrittr, shiny

**NeedsCompilation** no

**Author** Katharina Hees [aut],  
Peter Straka [aut, cre]

**Repository** CRAN

**Date/Publication** 2018-05-07 13:30:03 UTC

## R topics documented:

CTRE-package	2
acf	3
acf.ctre	4

bitcoin . . . . .	4
ctre . . . . .	5
empcopula . . . . .	6
flares . . . . .	6
interarrival . . . . .	7
length.ctre . . . . .	7
magnitudes . . . . .	8
MLEstimates . . . . .	8
mlqqplot . . . . .	9
plot.ctre . . . . .	10
qqestplot_static . . . . .	11
runCTREshiny . . . . .	11
seaquakes . . . . .	12
thin . . . . .	12
time . . . . .	13
<b>Index</b>	<b>14</b>

---

CTRE-package	<i>Continuous Time Random Exceedances</i>
--------------	---

---

## Description

Fits a Continuous Time Random Exceedance model to bursty time series. CTREs model the distribution of threshold crossing times of extreme events. Fits are made by interactive visual assessment of "stability plots" using a shiny app.

## References

"Peaks Over Threshold for Bursty Time Series", Katharina Hees, Smarak Nayak, Peter Straka.  
<https://arxiv.org/abs/1802.05218>

## Author(s)

**Maintainer:** Peter Straka <p.straka@unsw.edu.au>

Authors:

- Katharina Hees <hees@statistik.uni-dortmund.de>

## See Also

Useful links:

- <https://unsw-math.github.io/CTRE/>
- Report bugs at <https://github.com/UNSW-MATH/CTRE/issues>

**Examples**

```
library(CTRE)
## Not run:
runCTREshiny()

## End(Not run)

ctre_mod <- ctre(flare)
ctre_mod
length(ctre_mod)
ctre_mod <- thin(ctre_mod, k = 500)
ctre_mod
magnitudes(ctre_mod)
time(ctre_mod)
interarrival(ctre_mod)
MLEstimates(ctre_mod, tail = 0.8, scale = 3E7)
library(magrittr)
ctre_mod %>% interarrival() %>% mlqqplot(tail = 0.8, log = 'xy')
acf(ctre_mod)
empcopula(ctre_mod)
```

---

acf

*Autocorrelation function*

---

**Description**

Autocorrelation function

**Usage**

```
acf(x, ...)
```

**Arguments**

x                    time series or ctre object.  
...                   Additional arguments passed to `stats::acf`

**See Also**

[acf.ctre](#)

---

acf.ctre                      *Autocorrelation function*

---

### Description

Calculates and plots the autocorrelation function for the bivariate time series of interarrival times and magnitudes.

### Usage

```
## S3 method for class 'ctre'
acf(x, OCTRE = FALSE, ...)
```

### Arguments

x	An object of class <code>ctre</code>
OCTRE	If FALSE (default), each magnitude is matched with its preceding interarrival time. If TRUE, each magnitude is matched with its succeeding interarrival time.
...	Additional arguments passed to <code>acf</code>

### Examples

```
library(magrittr)
flares %>% ctre() %>% thin(k=150) %>% acf()
```

---

bitcoin                      *Bitcoin trading data*

---

### Description

A time series of trading volume of bitcoins, measured in "Bitcoin days destroyed", from 2010-12-04 until 2016-07-07.

### Usage

```
bitcoin
```

### Format

An object of class zoo of length 2043.

### Source

<https://datamarket.com/data/set/4aq9/various-bitcoin-currency-statistics>

### See Also

Other bursty-data: [flares](#), [seaquakes](#)

---

ctre	<i>CTRE model</i>
------	-------------------

---

### Description

Creates an object of class 'ctre' (Continuous Time Random Maxima).

### Usage

```
ctre(x, y = NULL, drop.duplicate.times = FALSE)
```

### Arguments

x	Either <ul style="list-style-type: none"><li>• a data.frame with two columns, or</li><li>• a vector; then y must be a vector of same length, or</li><li>• a zoo object.</li></ul> The first component/column must contain the event times and be of class "numeric", "Date" or "POSIXct"; the second component/column must contain the event magnitudes and be of type "numeric".
y	If x is a vector, y needs to be supplied as a vector of same length.
drop.duplicate.times	If there are duplicate time stamps present, should duplicates be dropped?

### Value

An object of class 'ctre' based on a time series of magnitudes.

### See Also

[CTRE](#)

### Examples

```
times <- cumsum(MittagLeffler::rml(n = 1000, tail = 0.8, scale = 5))
magnitudes <- rexp(n = 1000)
sim_ctre <- ctre(times, magnitudes)
sim_ctre
plot(sim_ctre)

library(magrittr)
bitcoin_ctre <- bitcoin %>% ctre() %>% thin(k = 400)
plot(bitcoin_ctre, log = 'y')
```

---

empcopula                      *Plot empirical copula*

---

### Description

Plots the ranks of the magnitudes against the ranks of the preceding (or succeeding) interarrival times.

### Usage

```
empcopula(ctre, OCTRE = FALSE, ...)
```

### Arguments

ctre	A <code>ctre</code> object
OCTRE	Shall each magnitude be matched with the preceding interarrival time (FALSE) or the succeeding interarrival time (TRUE)?
...	Additional plotting arguments

### Examples

```
library(magrittr)
flares %>% ctre() %>% thin(k = 300) %>% empcopula(pch = '*')
```

---

flares                              *Solar flare data*

---

### Description

A time series of peak rates of solar eruptions.

### Usage

```
flares
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2488 rows and 2 columns.

### Source

The Hard X-Ray Burst Spectrometer (HXRBS) on the Solar Maximum Mission, 14 February 1980 - 2 December 1989. <https://umbra.nascom.nasa.gov/smm/hxrbs.html>

### See Also

Other bursty-data: [bitcoin](#), [seaquakes](#)

---

interarrival	<i>Get inter-arrival times</i>
--------------	--------------------------------

---

**Description**

Extract inter-arrival times of threshold crossings from a `ctre` object

**Usage**

```
interarrival(ctre)
```

**Arguments**

<code>ctre</code>	The underlying <code>ctre</code> object
-------------------	---

**Examples**

```
library(magrittr)
bitcoin %>% ctre() %>% thin(k=100) %>% interarrival
```

---

<code>length.ctre</code>	<i>Get length of underlying time series</i>
--------------------------	---

---

**Description**

Extract length of underlying time series.

**Usage**

```
## S3 method for class 'ctre'
length(x)
```

**Arguments**

<code>x</code>	The <code>ctre</code> object containing the time series
----------------	---

**Examples**

```
library(magrittr)
flares %>% ctre() %>% thin(k=200) %>% length()
```

---

magnitudes	<i>Extract event magnitudes</i>
------------	---------------------------------

---

**Description**

Return the event magnitudes of a `ctre` object.

**Usage**

```
magnitudes(ctre, ...)
```

**Arguments**

<code>ctre</code>	A <code>ctre</code> object
<code>...</code>	Additional arguments passed to future methods.

**Value**

A numeric vector of event magnitudes

**Examples**

```
library(magrittr)
bitcoin %>% ctre() %>% magnitudes()
```

---

MLEstimates	<i>Mittag-Leffler estimates for varying thresholds</i>
-------------	--

---

**Description**

For a range of thresholds, return the parameters of the Mittag-Leffler distribution fitted to the threshold exceedance times.

**Usage**

```
MLEstimates(ctre, plot_me = TRUE, tail = NULL, scale = NULL,
  ks = 5:length(ctre))
```



**Arguments**

ctre	A <code>ctre</code> object
plot_me	Should the estimates be plotted?
tail	Tail parameter of the Mittag-Leffler distribution, if known. Appears as a dashed line in the plot of the tail parameter estimates, and transforms the scale parameter estimates. If not known, scale parameter estimates are untransformed (tail is set to 1).
scale	Scale parameter of the Mittag-Leffler distribution, if known. Appears as a dashed line in the plot of scale parameter estimates.
ks	The values of <code>k</code> at for which estimates are computed. If e.g. <code>k=10</code> , then the threshold is set at the 10th order statistic (10th largest magnitude), and Mittag-Leffler parameter estimates are coputed for the threshold exceedance times. By default, all order statistics are used except the 5 largest, and the estimates are returned in a data frame.

**Details**

If `plot_me = TRUE`, the estimates are returned invisibly.

**Value**

A `data.frame` of Mittag-Leffler parameter estimates, one row for each threshold, which is returned invisibly unless `plot_me = FALSE`.

**Examples**

```
library(magrittr)
par(mfrow = c(1,2))
flares %>% ctre() %>% thin(k=1000) %>% Mlestimates(tail = 0.9, scale = 3E7)

bitcoin %>% ctre() %>% thin(k=500) %>% Mlestimates(tail = 0.9, scale = 2.5E3)
bitcoin %>% ctre() %>% thin(k=500) %>% Mlestimates(plot_me = FALSE) %>% str()
```

---

mlqqplot

*Mittag-Leffler QQ Plot*


---

**Description**

Generates a QQ plot for assessing the fit of a Mittag-Leffler distribution.

**Usage**

```
mlqqplot(x, tail = 1, scale = NULL, ...)
```

**Arguments**

x	A vector of data to be compared against the Mittag-Leffler distribution.
tail	Tail parameter of the Mittag-Leffler population. Default is 1, i.e. the exponential distribution.
scale	Scale parameter of the Mittag-Leffler population, if known.
...	Additional plotting arguments, e.g. log = 'xy'.

**Examples**

```
library(magrittr)
flares %>% ctre() %>% thin(k=200) %>% interarrival() %>% mlqqplot(tail = 1, log = 'xy')
flares %>% ctre() %>% thin(k=200) %>% interarrival() %>% mlqqplot(tail = 0.8, log = 'xy')

seaquakes %>% ctre() %>% thin(k=150) %>% interarrival() %>% mlqqplot(tail = 0.9, log = 'xy')
```

---

plot.ctre

*Plot a ctre object*


---

**Description**

Plot a ctre object

**Usage**

```
## S3 method for class 'ctre'
plot(x, p = 0.05, ...)
```

**Arguments**

x	The ctre object whose time series should be plotted
p	The fraction of magnitudes that exceed the threshold to be plotted
...	Additional plotting parameters

**Examples**

```
library(magrittr)
flares %>% ctre() %>% plot(p = 0.02, log = 'y')
```

---

qqestplot_static	<i>Static QQ Plot estimator</i>
------------------	---------------------------------

---

**Description**

Generates a static QQ plot for Pareto tail estimates

**Usage**

```
qqestplot_static(data, top_k = NULL, plot_me = TRUE, ...)
```

**Arguments**

data	A vector of data.
top_k	Only use the top_k largest values (in the tail) for the plot
plot_me	Should a plot be produced? If not, only the estimate is returned.
...	Additional plotting arguments

**Value**

An estimate of the Pareto tail exponent (invisible if plotted).

---

runCTREshiny	<i>Run a shiny app to explore a CTRE model fit</i>
--------------	--

---

**Description**

Explore two pre-loaded datasets, simulated data, or upload your own.

**Usage**

```
runCTREshiny()
```

---

seaquakes

*Coral Sea Earthquake Data*

---

**Description**

A time series of earthquake magnitudes measured within the bounding box [166°E, 167°E] x [13.5°S, 12.4°S]

**Usage**

seaquakes

**Format**

An object of class `data.frame` with 754 rows and 2 columns.

**Source**

<http://seismicportal.eu/>

**See Also**

Other bursty-data: [bitcoin](#), [flares](#)

---

thin

*Apply a higher threshold to a CTRE process*

---

**Description**

Drop all but the `k` largest observations.

**Usage**

```
thin(ctre, k)
```

**Arguments**

`ctre` A `ctre` object.  
`k` Discard all but the `k` largest magnitudes.

**Value**

A `ctre` object, with fewer (`k`) observations.

**Examples**

```
library(magrittr)
flares %>% ctre() %>% plot(log = 'y')
flares %>% ctre() %>% thin(k=500) %>% plot(log = 'y')
```

---

time

*Sampling Times of Time Series*

---

**Description**

Sampling Times of Time Series

Extract event times from a `ctre` object.

**Usage**

```
time(x, ...)
```

```
## S3 method for class 'ctre'
time(x, ...)
```

**Arguments**

<code>x</code>	A time series or <code>ctre</code> object
<code>...</code>	Extra arguments for future methods

**Value**

A numeric vector of event times, or a time series.

**See Also**

[time.ctre](#)

**Examples**

```
library(magrittr)
seaquakes %>% ctre() %>% thin(k=200) %>% time()
```

# Index

## \*Topic **datasets**

- bitcoin, [4](#)
- flares, [6](#)
- seaquakes, [12](#)

acf, [3](#), [3](#), [4](#)  
acf.ctre, [3](#), [4](#)

bitcoin, [4](#), [6](#), [12](#)

CTRE, [5](#)  
CTRE (CTRE-package), [2](#)  
ctre, [4](#), [5](#), [6](#), [8](#), [9](#), [12](#)  
CTRE-package, [2](#)

empcopula, [6](#)

flares, [4](#), [6](#), [12](#)

interarrival, [7](#)

length.ctre, [7](#)

magnitudes, [8](#)  
Mlestimates, [8](#)  
mlqqplot, [9](#)

plot.ctre, [10](#)

qqestplot\_static, [11](#)

runCTREshiny, [11](#)

seaquakes, [4](#), [6](#), [12](#)

thin, [12](#)  
time, [13](#)  
time.ctre, [13](#)