

# Package ‘threshold4GPD’

October 26, 2016

**Type** Package

**Title** Determine Threshold by Independent Increments Structure of MLE

**Version** 0.8

**Author** Warren Jin, Data61, CSIRO Australia and Mengchen Zu,  
the Australian National University

**Maintainer** Warren Jin <Warren.Jin@csiro.au>

**Description** This is a package to determine threshold in extreme value  
analysis. Independent Increments Structure of MLE is used  
in this package to analyze the reasonable threshold for data set.

**Imports** ismev, mgcv, ADGofTest, dgof

**License** GPL-3

**LazyData** TRUE

**RoxygenNote** 5.0.1

## R topics documented:

Data_Randomization . . . . .	2
E.Info.Mat . . . . .	2
Expl.diag . . . . .	3
Joint.MLE.Expl . . . . .	4
Joint.MLE.NHPP . . . . .	5
K.ForwardStop . . . . .	6
K.StrongStop . . . . .	6
KvsN . . . . .	7
NHPP.diag . . . . .	7
nhpp.nll . . . . .	9
norm.LRT . . . . .	10
Threshold_Generator_Density . . . . .	11
Threshold_Generator_Uniform . . . . .	11
Threshold_Randomization . . . . .	12
<b>Index</b>	<b>13</b>

---

Data_Randomization	<i>Data_Randomization</i>
--------------------	---------------------------

---

**Description**

add the randomization of data to reduce the limitation of minimum measured value

**Usage**

```
Data_Randomization(x, random_digit = 2, round_digit = 4, seed = 1)
```

**Arguments**

x	- data vector
random_digit	- the precision of data vector
round_digit	- the precision of return randomization data vector
seed	- presetting random seed

**Value**

RandomData - randomization data vector

**Examples**

```
x <- rnorm(100)
x <- sort(x)
x <- round(x*10)/10
Data_Randomization(x)
```

---

E.Info.Mat	<i>E.Info.Mat</i>
------------	-------------------

---

**Description**

Calculates the numerically-integrated expected information matrix for an NHPP with specified parameters

**Usage**

```
E.Info.Mat(theta, u, M)
```

**Arguments**

theta	- vector of parameters (mu,sigma, xi)
u	- threshold for NHPP
M	- number of superpositions or "blocks" / "years" the process corresponds to

**Value**

EIM - expected information matrix

Errors - vector of errors from the numerical integration of the 6 unique components

---

Expl.diag	<i>producing the exponential diagnostic plots</i>
-----------	---

---

**Description**

Function to produce diagnostic plots and test statistics for the rate / inverse rate parameter of the Exponential model

**Usage**

```
Expl.diag(x, u = NULL, k, q1, q2 = 1, nbs = 1000, alpha = 0.05,
  plots = c("LRT", "WN", "PS"), UseQuantiles = TRUE, param = "InvRate",
  pmar = c(5.5, 7, 3, 3), ...)
```

**Arguments**

x	- data vector
u	- optional; vector of candidate thresholds
k	- number of thresholds to consider (if u unspecified)
q1	- lowest quantile for the threshold sequence
q2	- upper quantile limit for the threshold sequence (q2 itself is not used as a threshold, but rather the uppermost)
nbs	- number of simulations used to assess the null distribution of the LRT, and produce the p-value
alpha	- significance level of the LRT
plots	- which plots to produce; "LRT"= likelihood ratio test, "WN" = white noise, "PS" = parameter stability
UseQuantiles	- logical; use quantiles as the thresholds in the plot?
param	- character specifying "InvRate" or "Rate" for either inverse rate parameter / rate parameter, respectively
pmar	- vector of length 4 giving the arguments for the plot margins in par(mar=c(*,*,*,*))
...	- other parameter

**Value**

MLE - MLEs from all thresholds  
 Cov - Joint asymptotic covariance matrix for xi  
 WN - values of the white noise process  
 LRT - values of the LT test statistic vs threshold  
 pval - p-value of the LR test  
 k - final number of thresholds used  
 thresh - threshold selected by the LR procedure  
 mle.u - MLE from selected threshold

**Examples**

```
library(mvtnorm)
xbvn<-rmvnorm(6000, sigma=matrix(c(1,0.7,0.7,1),2,2))

# Transform margins to exponential
xbvn.exp<- -log(1-pnorm(xbvn))
Expl.diag(apply(xbvn.exp,1,min), k=30, q1=0, param="Rate")
Expl.diag(apply(xbvn.exp,1,min), k=30, q1=0, param="InvRate")
```

Joint.MLE.Expl

*Joint.MLE.Expl***Description**

Calculates the MLEs of the rate parameter, and joint asymptotic covariance matrix of these MLEs over a range of thresholds as supplied by the user.

**Usage**

```
Joint.MLE.Expl(x, u = NULL, k, q1, q2 = 1, param)
```

**Arguments**

x	- vector of data
u	- vector of thresholds. If not supplied, then k thresholds between quantiles (q1, q2) will be used
k	- number of thresholds to consider if u not supplied
q1	- lowest quantile for the threshold sequence
q2	- upper quantile limit for the threshold sequence (q2 itself is not used as a threshold, but rather the uppermost threshold will be at the (q2-1/k) quantile)
param	- character specifying "InvRate" or "Rate" for either inverse rate parameter / rate parameter, respectively

**Value**

mle - vector of MLEs above the supplied thresholds  
 cov - joint asymptotic covariance matrix of these MLEs

---

Joint.MLE.NHPP	<i>joint maximum likelihood estimation Non-Homogeneous Poisson Process</i>
----------------	--

---

**Description**

the MLEs of the parameters (mu,sigma,xi), and joint asymptotic covariance matrix of these MLEs over a range of thresholds as supplied by the user.

**Usage**

```
Joint.MLE.NHPP(x, u = NULL, k, q1, q2 = 1, par, M, MLE_model = "median")
```

**Arguments**

x	- vector of data
u	- vector of thresholds. If not supplied, then k thresholds between quantiles (q1, q2) will be used
k	- number of thresholds to consider if u not supplied
q1	- lowest quantile for the threshold sequence
q2	- upper quantile limit for the threshold sequence (q2 itself is not used as a threshold, but rather the uppermost threshold will be at the (q2-1/k) quantile)
par	- starting values for the optimization
M	- number of superpositions or "blocks" / "years" the process corresponds to (affects estimation of mu, sigma, but these can be changed post-hoc to correspond to any number)
MLE_model	- which model of maximum likelihood estimation is used; "median"= use the median data of 7 maximum likelihood estimation (not accurate enough but suitable for most data vector) "min"= use the min data of 7 maximum likelihood estimation (most accurate but not suitable for some small data vector) "second_min"= use the second min data of 7 maximum likelihood estimation (almost same as "min")

**Value**

mle - matrix of MLEs above the supplied thresholds; columns are (mu, sigma, xi)  
 Cov.all - joint asymptotic covariance matrix of all MLEs  
 Cov.mu - joint asymptotic covariance matrix of MLEs for mu  
 Cov.sig - joint asymptotic covariance matrix of MLEs for sig  
 Cov.xi - joint asymptotic covariance matrix of MLEs for xi

---

K.ForwardStop

*K.ForwardStop*


---

**Description**

calculate the cutoff k based on ForwardStop

**Usage**

K.ForwardStop(pvalues, alpha = 0.05)

**Arguments**

pvalues            - the p-values for each threshold in the threshold list  
alpha               - the pre-specified significance level, usually take 0.01 or 0.05

**Value**

forward - if the given null hypothesis should be rejected. Take 0 as no rejection and 1 as make rejection

---

K.StrongStop

*K.StrongStop*


---

**Description**

calculate the cutoff k based on StrongStop

**Usage**

K.StrongStop(pvalues, alpha = 0.05)

**Arguments**

pvalues            - the p-values for each threshold in the threshold list  
alpha               - the pre-specified significance level, usually take 0.01 or 0.05

**Value**

strong - if the given null hypothesis should be rejected. Take 0 as no rejection and 1 as make rejection

---

KvsN	<i>KvsN</i>
------	-------------

---

**Description**

Function to determine an appropriate number of thresholds

**Usage**

```
KvsN(x, wetCuttingPoint = 0.1)
```

**Arguments**

x                               - data vector  
wetCuttingPoint               - the minimum precision which can be considered as "rain"

**Value**

ks - an appropriate number of thresholds

---

NHPP.diag	<i>producing the Non-Homogeneous Poisson Process diagnostic plots</i>
-----------	---

---

**Description**

Function to produce diagnostic plots and test statistics for the NHPP model

**Usage**

```
NHPP.diag(x, u = NULL, k, q1 = NULL, q2 = 1, par = NULL, M = NULL,
nbs = 1000, alpha = 0.05, plots = c("LRT", "WN", "PS", "LRTEST", "KS",
"AD"), UseQuantiles = TRUE, DEBUGing = FALSE, pmar = c(5.5, 7, 3, 3),
head = 5, MLE_model = NULL, use_Data_Randomization = FALSE,
use_Threshold_Randomization = FALSE, ...)
```

**Arguments**

x                               - data vector  
u                               - optional; vector of candidate thresholds  
k                               - number of thresholds to consider (if u unspecified)  
q1                               - lowest quantile for the threshold sequence  
q2                               - upper quantile limit for the threshold sequence (q2 itself is not used as a threshold, but rather the uppermost threshold will be at the (q2-1/k) quantile)

<code>par</code>	- starting values for the optimization
<code>M</code>	- number of superpositions or "blocks" / "years" the process corresponds to (can affect the optimization)
<code>nbs</code>	- number of simulations used to assess the null distribution of the LRT, and produce the p-value
<code>alpha</code>	- significance level of the LRT
<code>plots</code>	- which plots to produce; "LRT" = likelihood ratio "WN" = white noise "PS" = parameter stability "LRTEST" = likelihood ratio test for assistance "KS" = Kolmogorov-Smirnov goodness-of-fit test for assistance "AD" = Anderson-Darling test for assistance
<code>UseQuantiles</code>	- logical; use quantiles as the thresholds in the plot?
<code>DEBUGing</code>	- default as false, set to true if you want to debug
<code>pmar</code>	- vector of length 4 giving the arguments for the plot margins in <code>par(mar=c(*,*,*,*))</code>
<code>head</code>	- number of thresholds inserted between the first threshold and the second threshold temporary to reduce the variance at the beginning of likelihood ratio statistic test
<code>MLE_model</code>	- which model of maximum likelihood estimation is used; "median" = use the median data of 7 maximum likelihood estimation (not accurate enough but suitable for most data vector) "min" = use the min data of 7 maximum likelihood estimation (most accurate but not suitable for some small data vector) "second_min" = use the second min data of 7 maximum likelihood estimation (almost same as "min")
<code>use_Data_Randomization</code>	- whether or not use Data_Randomization to randomize the given data
<code>use_Threshold_Randomization</code>	- whether or not use Threshold_Randomization to randomize the thresholds
<code>...</code>	- other parameter

## Details

Function to produce diagnostic plots and test statistics for the NHPP model

## Value

MLEall - MLEs from all thresholds  
Cov.xi - Joint asymptotic covariance matrix for xi  
WN - values of the white noise process  
LRT - values of the LT test statistic vs threshold  
pval - p-value of the LR test  
k - final number of thresholds used  
thresh - threshold selected by LR procedure  
mle.u - MLE from selected threshold



## Examples

```
## insert an easy example for test run
set.seed(1)
xnorm<-abs(rnorm(5000))
thresholds_xnorm <- Threshold_Generator_Uniform(xnorm, wetCuttingPoint=0.1)
ChangedThresholds_xnorm <- Threshold_Randomization(xnorm,thresholds_xnorm,seed = 1)
nhpp <-NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("LRT", "WN", "PS"))

##insert an example for test run
set.seed(1)
x <- rnorm(100000)
x <- x[x > quantile(x, probs = 0.9)]
x <- sort(x)
x <- Data_Randomization(x,seed = 1)
x <- sort(x)
thresholds_x <- Threshold_Generator_Uniform(x, wetCuttingPoint=0.1)
ChangedThresholds_x <- Threshold_Randomization(x,thresholds_x,seed = 1)
nhpp <-NHPP.diag(x, u= ChangedThresholds_x,
                 M=365, nbs=1000, alpha=0.05, plots=c("LRT", "WN", "PS"), UseQuantiles=FALSE,
                 cex.lab=1.5, cex.axis=1.4,cex.main=2,mgp=c(4.2,1,0))

#####
# View different plots for easy example:
# likelihood ratio plot only
NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("LRT"))

# white noise plot only
NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("WN"))

# parameter stability plot only
nhpp <-NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("PS"))

# likelihood ratio test for assistance plot only
NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("LRTEST"))

# Kolmogorov-Smirnov goodness-of-fit test for assistance plot only
NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("KS"))

# Anderson-Darling test for assistance plot only
NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("AD"))

# a showoff of all the plots
NHPP.diag(xnorm, u= ChangedThresholds_xnorm, plots=c("LRT", "WN", "PS", "LRTEST", "KS", "AD"))
```

## Description

Negative log likelihood for the NHPP model, to minimize for MLEs

**Usage**

`nhpp.nll(theta, x, u, M)`

**Arguments**

- `theta` - parameter vector ( $\mu$ ,  $\sigma$ ,  $\xi$ )
- `x` - data vector
- `u` - threshold
- `M` - number of superpositions or "blocks" / "years" the process corresponds to (affects estimation of  $\mu$ ,  $\sigma$ , but these can be changed post-hoc to correspond to any number)

**Value**

negative log-likelihood value

---

<code>norm.LRT</code>	<i>normal distribution likelihood ratio statistics</i>
-----------------------	--

---

**Description**

Evaluates the likelihood ratio statistics for testing white noise

**Usage**

`norm.LRT(x, u)`

**Arguments**

- `x` - vector of white noise process (WNP, usually normalized estimates of  $\xi$  or the exponential rate parameter  $1/\eta$ )
- `u` - vector of thresholds that are associated to the WNP

**Value**

`v`  
likelihood ratio

---

Threshold\_Generator\_Density  
*Thershold\_Generator\_Density*

---

### Description

generate a candidate thresholds list for given data set based on variant distance among the distribution density

### Usage

```
Threshold_Generator_Density(x, wetCuttingPoint = 0.1, ks = NULL, q = NULL)
```

### Arguments

x	- data vector
wetCuttingPoint	- the minimum precision which can be considered as "rain"
ks	- number of candidate thresholds
q	- presetting value to control the density (usually set between 1 and 2) (the bigger q is, the more intensive thresholds in tail part)

### Value

thresholds - candidate thresholds list

### Examples

```
set.seed(1)
xnorm<-abs(rnorm(5000))
Threshold_Generator_Density(xnorm)
```

---

Threshold\_Generator\_Uniform  
*Thershold\_Generator\_Uniform*

---

### Description

generate a candidate thresholds list for given data set based on uniform distance among the distribution

### Usage

```
Threshold_Generator_Uniform(x, wetCuttingPoint = 0.1, ks = NULL)
```

**Arguments**

x - data vector  
wetCuttingPoint - the minimum precision which can be considered as "rain"  
ks - number of candidate thresholds

**Value**

thresholds - candidate thresholds list

**Examples**

```
set.seed(1)
xnorm<-abs(rnorm(5000))
Threshold_Generator_Uniform(xnorm)
```

---

Threshold\_Randomization

*Threshold\_Randomization*

---

**Description**

add the randomization of thresholds to increase the randomization of thresholds

**Usage**

```
Threshold_Randomization(x, thresholds, seed = 1)
```

**Arguments**

x - data vector  
thresholds - thresholds vector  
seed - presetting random seed

**Value**

thresholds - randomization thresholds vector

**Examples**

```
set.seed(1)
xnorm<-abs(rnorm(5000))
thresholds_xnorm <- Threshold_Generator_Uniform(xnorm)
thresholds_xnorm
Threshold_Randomization(xnorm,thresholds_xnorm,seed = 1)
```

# Index

Data\_Randomization, [2](#)

E.Info.Mat, [2](#)

Expl.diag, [3](#)

Joint.MLE.Expl, [4](#)

Joint.MLE.NHPP, [5](#)

K.ForwardStop, [6](#)

K.StrongStop, [6](#)

KvsN, [7](#)

NHPP.diag, [7](#)

nhpp.nll, [9](#)

norm.LRT, [10](#)

Threshold\_Generator\_Density, [11](#)

Threshold\_Generator\_Uniform, [11](#)

Threshold\_Randomization, [12](#)