Package ‘Renext’
February 15, 2013

Type Package
Title Renewal method for extreme values extrapolation
Version 2.0-11
Date 2012-08-23
Author Yves Deville <deville.yves@alpestat.com>, IRSN <renext@irsn.fr>
Maintainer Lise Bardet <lise.bardet@irsn.fr>
Depends R (>= 2.8.0), evd, numDeriv
Suggests MASS, graphics, ismev, XML
Description R package dedicated to some Extreme values problems and allowing the use of the so-called ‘renewal method’ which is popular among french-speaking hydrologists.
License GPL (>= 2)
LazyLoad yes
URL https://gforge.irsn.fr/gf/project/renext
Repository CRAN
Date/Publication 2012-08-24 12:33:23
NeedsCompilation no

R topics documented:

Renext-package ................................................................. 2
barplotRenouv ............................................................... 4
Brest ................................................................. 7
Brest.years ................................................................. 8
Brest.years.missing ....................................................... 9
Dunkerque ................................................................. 10
Renext-package

Renewal method for extreme values extrapolation

Description

This package proposes fits and diagnostics for the so-called "renewal method", an alternative to other "Peaks Over Threshold" (POT) methods. The renewal method generalises the classical POT by allowing the exceedances over the threshold to follow a probability distribution which can differ from the Generalised Pareto Distribution (GPD). Weibull or gamma exceedances are sometimes preferred to GPD exceedances. The special case of exponential exceedances (which falls in the three families: GPD, Weibull and gamma) has a special interest since it allows exact inference for the (one dimensional) parameter and for the quantiles form OT data (only).

The package allows the joint use of possibly three kinds of data or information. The first kind is classical exceedances, or "OT data" and will always be required. It can be completed with two kinds of historical data. Both types are optional, and concern periods or blocks that must correspond to non-overlapping periods, disjoint from the OT period.
• *MAX data* correspond to the case where one knows the \( r \) largest observations over each block. The number \( r \) of may vary between blocks. This kind of data is often called \( r \)-max data.

• *OTS data* (for OT Supplementary data) correspond to the case where one knows for each block \( b \) all the observations that exceeded a threshold \( u_b \) which is greater (usually much greater) than the main threshold \( u \). The number \( r_b \) of such observations can be zero, in which case we may say that \( u_b \) is an unobserved level.

Historical data are often available in hydrology (e.g. for river flows) for large periods such as past centuries. An unobserved level can typically be related to a material benchmark.

Maximum likelihood estimation is made possible in this context of heterogeneous data. Inference is based on the asymptotic normality of parameter vector estimate and on linearisation ("delta method") for quantiles or parameter functions.

The package allows the use of "marked-process observations" data (datetime of event and level) where an interevent analysis can be useful. It also allows that the event dates are unknown and replaced by a much broader block indication, e.g. a year number. The key point is then that the "effective duration" (total duration of observation periods) is known. Event counts for blocks can be used to check the assumption of Poisson-distributed events.

The package development was initiated, directed and financed by the french *Institut de Radioprotection et de Sûreté Nucléaire* (IRSN). The package is a non-academic tool designed for applied analysis on case studies and investigations or comparisons on classical probabilistic models.

Additional information and material related to this package can be found at the URL [https://gforge.irsn.fr/gf/project/renext](https://gforge.irsn.fr/gf/project/renext).

**Details**

<table>
<thead>
<tr>
<th>Package:</th>
<th>Renext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>2.0-11</td>
</tr>
<tr>
<td>Date:</td>
<td>2012-08-23</td>
</tr>
<tr>
<td>License:</td>
<td>GPL (&gt;= 2)</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

This package contains a function Renouv to fit "renouvellement" models.

**Author(s)**

Yves Deville <deville.yves@alpestat.com>, IRSN <renext@irsn.fr>

Maintainer: Lise Bardet <lise.bardet@irsn.fr>

**References**


See Also

The packages evd, ismev, extRemes, bayesevd, POT.

Examples

```r
## Garonne data set
data(Garonne)
summary(Garonne)
plot(Garonne)

## Weibull exceedances
fG <- Renouv(x = Garonne,
             threshold = 3000,
             distname.y = "weibull",
             main = "Weibull fit for 'Garonne'")

coef(fG)
summary(fG)
## classical 'predict' method with usual formal args
predict(fG, newdata = c(100, 150, 200),
        level = c(0.8, 0.9))
```

barplotRenouv

Barplot for Renouv "Over Threshold" counts

Description

Barplot for "Over Threshold" counts in time blocks (usually years)

Usage

```r
barplotRenouv(data,
              blockname = colnames(data)[1],
              varname = colnames(data)[2],
              threshold = quantile(data[, varname], 0.2),
              na.block = NULL,
              plot = TRUE,
              main = NULL, xlab = NULL, ylab = NULL,
              mono = FALSE,
              prob.theo = 0.999,
              ...)```

Arguments

data A dataframe object containing the variables.
blockname Name of the "block" variable (column in data). This variable should contain integers, or be of class "factor", but with integer values such as year numbers.
varname Name of the variable (e.g. "Surge").
threshold      Only obs for which the variable exceeds threshold will be taken into account.
na.block      Values of blocks containing missing values. See the Details section.
plot          If FALSE tests are computed without producing any plot.
main          Character for main title or NULL in which case a default main title is used.
xlab          Character for x axis label or NULL in which case a default lab is used.
ylab          Character for y axis or NULL in which case a default lab is used.
mono          If FALSE barplot will have colors, else grayscale will be used.
prob.theo     The total theoretical probability corresponding to the plotted (theoretical) bars.
...
              Further args to be passed to barplot.

Details
Blocks described in the na.block are omitted in the determination of counts. The object given in the na.block is coerced to character and the same is done for values of block before comparing them to the na.block values. If block variable is of class factor with levels representing years (e.g. 1980, 1981, etc.) missing blocks can be specified either as c("198/zero.noslash", "1981") or as numeric c(198/zero.noslash, 1981).
For the chi-square test, counts for neighbouring frequency classes are grouped in order to reach a minimum frequency of 5 in each group. E.g. if we expect respectively 1.0, 3.8 and 7.0 blocks with frequency 0, 1 and 2 for events, the three counts are grouped in one group with frequency 1.0+3.8+7.0=11.8. Note that this strategy of grouping is not unique and is likely to weaken the power of the test. Before grouping, the higher class theretical probability is computed as the probability to obtain a count equal to or greater than the max value.

Value
A list with the following objects.

freq          frequency table (matrix) giving observed and theoretical (Poisson) frequencies as well as a group number for the chi-square test.
overdispersion the overdispersion coefficient (variance/mean ratio).
disp.test      a list giving results of the (over)dispersion test. See the reference Yagouti and al. in the References section.
chisq.test     a list giving results for the chisquare test of goodness-of-fit to the Poisson distribution.
tests         a matrix with the two tests displayed in two rows.

For both tests, the statistic follows a chi-square distribution under the null hypothesis . The list of results contains the statistic statistic, the number of degrees of freedom df and the p-value p.value.

Note
The two tests: (over-)dispersion and chi-square have one-sided (upper tail) p-value. In other words, we do not intend to reject when statistics take "abnormally small" values, but only when abnormally large values are met.
Author(s)

Yves Deville

References


See Also

plot.Rendata

Examples

library(Renext)
data(Brest.years)
data(Brest.years.missing)

## na.block influence
opar <- par(mfrow = c(2, 2))

bp1 <- barplotRenouv(data = Brest.years, threshold = 3/zero.noslash,
  main = "missing periods ignored")
bp2 <- barplotRenouv(data = Brest.years, threshold = 3/zero.noslash,
  na.block = 1992, main = "1992 missing")
bp3 <- barplotRenouv(data = Brest.years, threshold = 3/zero.noslash,
bp4 <- barplotRenouv(data = Brest.years, threshold = 3/zero.noslash,
  na.block = Brest.years.missing, main = "all missing periods")

par(opar)

## threshold influence
opar <- par(mfrow = c(2,2))

thresh <- c(30, 35, 40, 50)

for (i in 1:length(thresh)) {
  bp <- barplotRenouv(data = Brest.years, threshold = thresh[i],
    na.block = Brest.years.missing,
    main = paste("threshold =", thresh[i], " cm at Brest"))
}

par(opar)
**Brest**

**Surge heights at Brest**

**Description**

Surge heights near high tide at Brest tide gauge station (France), detailed version

**Usage**

data(Brest)

**Format**

The format is: List of 5

- `$info`: List of 6
  - `$name`: chr "Brest"
  - `$shortLab`: chr "Surge Heights at Bres (France)"
  - `$longLab`: chr "Surge Heights near high tide, Brest (France)"
  - `$varName`: chr "Surge"
  - `$varShortLab`: chr "Surge"
  - `$varUnit`: chr "cm"

- `$describe`: chr "High tide sea surge over 30 cm at Brest (France) separated by at least two days."

- `$OTinfo`: List of 4
  - `$start`: chr POSIXct[1:1], format: "1845-12-31 23:59:39"
  - `$end`: chr POSIXct[1:1], format: "2009-01-01"
  - `$effDuration`: num 148
  - `$threshold`: num 30

- `$OTdata`: 'data.frame': 1289 obs. of 2 variables:
    ...
  - `$Surge`: num [1:1289] 36 60 46 43 ...

- `$OTmissing`: 'data.frame': 43 obs. of 3 variables:
    ...
    ...
  - `$comment`: chr [1:43] NA NA NA NA ...

- attr(*, "class")= chr "Rendata"
Details

Data are provided as a list.

- **info** gives general information about the data
- **OTinfo** gives general information about the Over the Threshold part of data. The effective duration (effduration element) is the total duration for the periods with effective measurements.
- **OTdata** give OT measurements
- **OTmissing** gives start and end of the missing periods for OT measurements.

Data come from hourly sea levels measured and predicted by the French *Service Hydrogéographique et Océanographique de la Marine* (SHOM). Observed sea levels are available at the url [http://www.sonel.org](http://www.sonel.org). Data were processed (declustered) by IRSN in order to provide a series of independent surge heights at high tide. Surge height at high tide is defined as the difference between the observed and the predicted maximal sea levels near high tide. A correction was applied to account for trend in the sea-level over the observation period.

The effective duration given in years is defined up to a small fraction of year due to leap years and leap seconds.

Source

[http://www.sonel.org](http://www.sonel.org)

Examples

```r
data(Brest)
str(Brest)
Brest$OTinfo$start
plot(Brest)
```

---

**Brest.years** | Surge heights at Brest partial data

Description

Surge heights at Brest (France)

Usage

```r
data(Brest.years)
```

Format

A data frame with 954 observations on the following 2 variables.

- **year** Year e.g. 1980
- **Surge** Surge heights above the threshold of 30 cm.
These data are a simplified version of Brest. For each surge event only the year is retained as timestamp. Years with missing periods are available as a vector Brest.years.missing.

This dataset is useful for testing since similar data are sometimes met in the analyses.

**Examples**

```r
data(Brest.years)
names(Brest.years)
```

---

**Brest.years.missing**  
*Years with missing periods in 'Brest.year' dataset*

---

**Description**

Years with missing periods in the 'Brest.years' dataset

**Usage**

```r
data(Brest.years.missing)
```

**Format**

The format is: int [1:49] 1846 1847 1852 1857 1858 1859 1860 1861 1862 1863 ...

**Details**

Vector of years containing missing periods in the Brest.years dataset. This years should be ignored when computing yearly statistics such as event rates, since time records are lost.

**Examples**

```r
data(Brest.years.missing)
print(Brest.years.missing)
```
Dunkerque

Surge heights at Dunkerque

Description

Surge heights near high tide at Dunkerque tide gauge station (France)

Usage

data(Dunkerque)

Format

The format is: List of 7

- **$info**: List of 6
  - $name: chr "Dunkerque"
  - $shortLab: chr "Surge Heights at Dunkerque (France)"
  - $longLab: chr "Surge Heights near high tide, Dunkerque (France)"
  - $varName: chr "Surge"
  - $varShortLab: chr "Surge"
  - $varUnit: chr "cm"
- **$describe**: chr "High tide sea surge over 30 cm at Dunkerque (France) separated by at least two days."
- **$OTinfo**: List of 4
  - $start: POSIXct[1:1], format: "1956-01-01"
  - $end: POSIXct[1:1], format: "2009-01-01"
  - $effDuration: num 38.8
  - $threshold: num "30"
- **$OTdata**: 'data.frame': 740 obs. of 3 variables:
  - $date: POSIXct[1:740], format: "1956-11-27" "1956-12-03" ...
  - $Surge: num [1:740] 67.9 30.9 51.8 30.8 39.8 ...
  - $comment: Class 'AsIs' chr [1:740] "" "" "" "" ...
- **$OTmissing**: 'data.frame': 83 obs. of 3 variables:
  - $start: POSIXct[1:83], format: "1956-01-01" "1956-08-08" ...
  - $end: POSIXct[1:83], format: "1956-06-07" "1956-11-03" ...
  - $comment: Class 'AsIs' chr [1:83] "" "" "" "" ...
- **$MAXinfo**: 'data.frame': 1 obs. of 3 variables:
  - $start: POSIXct[1:1], format: "1956-01-01"
  - $end: POSIXct[1:1], format: "1956-01-01"
  - $duration: num 250
- **$MAXdata**: 'data.frame': 1 obs. of 4 variables:
Details

See Brest and Garonne datasets with the same list structure.

An 'historical' surge of 213 cm was observed on 1953-02-01 and is considered by experts as having a return period of 250 years.

Examples

data(Dunkerque)
Dunkerque$info
plot(Dunkerque)

EM.mixexp  Expectation-Maximisation for a mixture of exponential distributions

Description

Experimental function for Expectation-Maximisation (EM) estimation

Usage

EM.mixexp(x, m = 2)

Arguments

x  Sample vector with values >0.

m  Number of mixture components

Details

The EM algorithm is very simple for exponential mixtures (as well as for many other mixture models).

According to a general feature of EM, this iterative method leads to successive estimates with increasing likelihood but which may converge to a local maximum of the likelihood.
Value

- **estimate**: Estimated values as a named vector.
- **logL**: Vector giving the log-likelihood for successive iterations.
- **Alpha**: Matrix with \( m \) columns giving probability weights for successive iterations. Row with number \( it \) contains the \( m \) probabilities at iteration \( it \).
- **Theta**: Matrix with \( m \) columns giving the estimates of the \( m \) expectations for the successive iterations.

Note

The estimation is done for expectation (inverse rates) but the estimate vector in the result contains rates for compatibility reasons (e.g. with exponential).

Author(s)

Yves Deville

See Also

- `mom.mixexp2` and `ini.mixexp2` for "cheap" estimators when \( m = 2 \).

Examples

```r
library(Renext)
set.seed(1234)
x <- rmixexp2(n = 100, prob1 = 0.5, rate2 = 4)
EM.mixexp(x) -> res
res$estimate
matplot(res$Theta, type = "l", lwd = 2,
xlab = "iteration", ylab = "theta",
main = "exponential inverse rates")
```

Description

Plot a vector using "exponential distribution" scales

Usage

```r
expplot(x,
plot.pos = "exp",
rate = NULL,
labels = NULL,
mono = TRUE,
...)
```
expplot

**Arguments**

- `x` The vector to be plotted
- `plot.pos` Plotting position for points: either "exp" for expected ranks or "med" for a median rank approximation (see Details below).
- `rate` Rate parameter for one or several "exponential distribution" lines to be plotted
- `labels` Text to display in legend when "exponential distribution" lines are specified
- `mono` Monochrome graph?
- `...` Arguments to be passed to plot

**Details**

This plot shows $-\log(1 - F(x))$ against $x$ where $F(x)$ at point $i$ is taken as $i/(n + 1)$ if `plot.pos` is "exp", or as the "median rank" approximation $(i - 0.3)/(n + 0.4)$ if `plot.pos` is "med".

If the data in `x` is a sample from an exponential distribution, the points should be roughly aligned. However the largest order statistics have high sampling dispersion.

**Note**

The log scale for y is emulated via the construction of suitable graduations. So be careful when adding graphical material (points, etc) to this graph with functions of the "add to plot" family (`points`, `lines`, ...).

The ML estimate of the `rate` parameter is the inverse of the sample mean.

**Author(s)**

Y. Deville

**See Also**

The `weibplot` function for a classical "Weibull" plot. The `interevt` is useful to compute interevents (or "interarrivals") that should follow an exponential distribution in the homogeneous Poisson process context.

**Examples**

```r
x <- rexp(200)
expplot(x, rate = 1/mean(x), labels = "fitted")
```
fweibull

ML estimation of classical Weibull distribution

Description

Fast Maximum Likelihood estimation of the classical two parameters Weibull distribution

Usage

fweibull(x,
    info.observed = FALSE,
    check.loglik = FALSE)

Arguments

x Sample vector to be fitted
info.observed Should the observed information matrix be used or the expected one be used?
check.loglik If TRUE, the log-likelihood is recomputed using dweibull function with log = TRUE. The result is returned as a list element.

Details

The ML estimates are obtained thanks to a reparametrisation with $\eta = \text{scale}^{1/\text{shape}}$ in place of shape. This allows the maximisation of a one-dimensional likelihood $L$ since the $\eta$ parameter can be concentrated out of $L$. This also allows the determination of the expected information matrix for $[\text{shape}, \eta]$ rather than the usual observed information.

Value

A list

- estimate parameter ML estimates
- sd (asymptotic) standard deviation for estimate
- cov (asymptotic) covariance matrix computed from theoretical or observed information matrix
- eta the estimated value for eta

Note

A well-accepted verdict is that expected information is better than observed information for problems with no missing data as it is the case here. We suspect in the present estimation context that the expected information matrix has often a better condition number than its observed version.

Author(s)

Yves Deville
See Also

*weibplot* for Weibull plots

Examples

```r
library(MASS)

n <- 1000
shape <- 2*runif(1)
x <- 100*rweibull(n, shape = 0.8, scale = 1)
res <- fweibull(x)
res2 <- fitdistr(x, "weibull")
est <- cbind(res$estimate, res2$estimate)
colnames(est) <- c("Renext", "MASS")
loglik <- c(res$loglik, res2$loglik)
est <- rbind(est, loglik)
est

weibplot(x,
  shape = c(res$estimate["shape"], res2$estimate["shape"]),
  scale = c(res$estimate["scale"], res2$estimate["scale"]),
  labels = c("Renext 'fweibull'", "MASS 'fitdistr'"),
  mono = TRUE)
```

---

### Garonne

**Flow of the french river La Garonne**

---

**Description**

Flow of the french river La Garonne at le Mas d'Agenais

**Usage**

data(Garonne)

**Format**

The format is: List of 7

- `$info`: List of 6
  - `$name`: chr "Garonne"
  - `$shortLab`: chr "La Garonne at Le Mas d’Agenais"
  - `$longLab`: chr "River flow of La Garonne at Le Mas d’Agenais"
  - `$varName`: chr "Flow"
  - `$varShortLab`: chr "Flow"
  - `$varUnit`: chr "m3/s"
- `$describe`: chr "Flow of the french river La Garonne at the gauging station Le Mas d’Agenais. The data are taken from Miquel’s book."
- `$OTinfo`: List of 4
The data concern the French river La Garonne at the gauging station named Le Mas d'Agenais where many floods occurred during the past centuries.

The data consist in OT data and historical data. The variable is the river flow in cubic meters per second (m³/s) as estimated from the river level using a rating curve. The precision is limited and many ties are present among the flow values.

The OT data or "OTdata" contain flows values over the threshold \( u = 2500 \) m³/s for the 65 years period 1913-1977. The historical data or "MAXdata" is simply the \( r = 12 \) largest flows for the period of 143 years 1770-1912. The exact dates of these events are not known with precision but the years are known and given as comments.

Source

The data were taken from the book by Miquel.

References


Examples

data(Garonne)
plot(Garonne)
gof.date

Goodness-of-fit for the repartition of dates

Description

Goodness-of-fit diagnostics for the repartition of event dates in a (assumed) Poisson process

Usage

```r
gof.date(date, 
  start = NULL, 
  end = NULL, 
  plot = TRUE, 
  main = NULL, 
  skip = NULL, 
  plot.type = "skip")
```

Arguments

- **date**: Object of class POSIXct (or that can be coerced to this class) giving the dates to be tested. Must be in strictly increasing order.
- **start**: The beginning of the interval, a POSIXct object. If NULL, the first event in date is used.
- **end**: Object of class POSIXct the end of the interval. If NULL, the last event in date is used.
- **plot**: Should a plot be shown?
- **main**: Character giving the main title of the plot. The default NULL stands for a default main describing the period.
- **skip**: Optional data.frame with columns `start` and `end` indicating start and end of skipped periods. The two columns need to be coerced to POSIXct objects. They can be POSIXct or character with POSIX datetime format.
- **plot.type**: Character indicating the type of plot to produce when a `skip` data.frame is given. With `plot.type = "skip"` the plot shows missing periods as greyed rectangles and the displays the results of a Kolmogorov-Smirnov (KS) test performed on the events. For the "omit" case the missing periods are collapsed into vertical lines on the plot and the displayed results are for an "effective" KS test of uniformity performed omitting the missing periods.

Details

In the homogeneous Poisson process, events occur on a time interval in a uniform fashion. More precisely, for a given time interval the distribution of the event dates conditional to their number \( n \) is the distribution of the order statistics of a sample of size \( n \) of the uniform distribution on this interval.

When the interval has limits taken at events the uniformity statement remains true, but for inner events. This behaviour is met when `start` and `end` are not given and taken as the first and last events in `date`. 
Value

A list

- effKS.statistic, KS.statistic
  Kolmogorov-Smirnov global test statistic for uniformity (bilateral test) omitting
  slipped periods or not.

- effKS.pvalue, KS.pvalue
  Critical probability in the KS test omitting skipped periods or not.

- effnevt, nevt
  Number of events omitting skipped periods or not.

- effduration, duration
  Effective duration i.e. total duration of non-skipped periods. In years, omitting
  skipped periods or not.

- effrate, rate
  Occurrence rate in number of events by year, omitting skipped periods or not.

- effduration, duation
  Total duration in years, omitting missing periods or not.

- noskip
  Data.frame object giving indications on the periods that are NOT skipped over
  (hence usually non-missing periods). These are: start, end (POSIX), duration
  (in years) rate (in number of events by year) and Kolmogorov test statistic and
  p-value. This data.frame is only available when a suitable skip has been given.

When the number of events corresponding to the indications of args is 0, the function returns NULL
with a warning. When the number of events is less than 6 a warning is shown.

Warning

When skipped periods exist the number of events, duration, rate the global KS test must be com-
puted by omitting the skipped periods in the duration and retaining only valid interevents. The
indication given in nevt rate and duration should be used only when no skipped period exist
(skip = NULL on input) and replaced by effnevt, effrate and effduration otherwise.

Note

In practical contexts missing periods are often met in the datasets. The diagnostic should therefore
be applied on every period with no missing data. Even if the event dates seem reasonably uniform,
it is a good idea to check that the rates do not differ significantly over intervals.

When some events are missing and no suitable information is given via the skip argument, the
global rate, KS.statistic and KS.pvalue are of little interest. Yet the graph might be instructive.

Author(s)

Yves Deville

See Also

interevt function for the determination of interevents ans subsequent diagnostics.
Examples

```r
## Use "Brest" dataset
data(Brest)

## simple plot. Kolmogorov-Smirnov is not useful
gof1 <- gof.date(date = Brest$OTdata$date)

## consider missing periods. Much better!
gof2 <- gof.date(date = Brest$OTdata$date,
                  skip = Brest$OTmissing,
                  start = Brest$OTinfo$start,
                  end = Brest$OTinfo$end)
print(gof2$noskip)

## Second type of graph
gof3 <- gof.date(date = Brest$OTdata$date,
                  skip = Brest$OTmissing,
                  start = Brest$OTinfo$start,
                  end = Brest$OTinfo$end,
                  plot.type = "omit")

## non-skipped periods at Brest
ns <- skip2noskip(skip = Brest$OTmissing,
                   start = Brest$OTinfo$start,
                   end = Brest$OTinfo$end)

## say 9 plots/diagnostics
oldpar <- par(mar = c(3, 4, 3, 2), mfcol = c(3, 3))
for (i in 1:9) {
  GOF <- gof.date(date = Brest$OTdata$date,
                  start = ns$start[i],
                  end = ns$end[i])
}
par(oldpar)
```

---

gofExp.test  
*Goodness-of-fit test for exponential distribution*

Description

Bartlett’s goodness-of-fit test for exponential distribution

Usage

gofExp.test(x)
Arguments

x sample

Value

A list with elements

statistic Statistic
p.value   Critical value

Author(s)

Yves Deville

References


See Also

Among other goodness-of-fit tests ks.test in the stats package. See expplot for a graphical diagnostic.

Examples

## a sample of size 30
x <- rexp(30)
res <- gofExp.test(x)

## ns samples: p.values should look as uniform on (0, 1)
ns <- 100
xmat <- matrix(rexp(30*ns), nrow = ns, ncol = 30)
p.values <- apply(xmat, 1, function(x) gofExp.test(x)$p.value)
plot(sort(p.values), type = "p", pch = 16)

ini.mixexp2

Simple estimation for the mixture of two exponential distributions

Description

Compute a simple (preliminary) estimation for the tree parameters of the mixture of two exponential distributions

Usage

ini.mixexp2(x, plot = FALSE)
Arguments

x        Sample: numerical vector with elements >0.
plot     Should a graphic be displayed?

Details

This function gives estimators using several methods if necessary. The goal is to find the rates rate1, rate2 and the mixing probability prob1 with the 'feasibility' constraints 0 < rate1 < rate2 and 0 < prob1 < 1.

First the method of moments is used. If the estimates are feasible they are returned with method = "moments". If not, the estimates are derived using two linear regressions. A regression without constant using only the smallest values gives an estimator of the mean rate. A regression using only the largest values gives rate1 and prob1. Yet the constraints must be fulfilled. If they are, the estimates are returned (together with method = "Hreg" suggesting a cumulative hazard regression). If not, a (poor) default estimate is returned with method = "arbitrary".

Value

A list

estimate A vector with named elements "prob1", "rate1" and "rate2".
method   The method that really produced the estimators.

Note

The method of moments is implemented in mom.mixexp2. Further investigations are needed to compare the estimators (moment or Hreg) and select the best strategy.

Note that this function returns the estimate within a list and no longer as a vector with named elements as was the case before.

Author(s)

Yves Deville

See Also

See MixExp2, mom.mixexp2.

Examples

library(Renext)
set.seed(1234)
x <- rmixexp2(n = 100, prob1 = 0.5, rate2 = 4)
res <- ini.mixexp2(x, plot = TRUE)
**Description**

Compute interevent durations from events dates

**Usage**

```r
interevt(date,
    skip = NULL, noskip = NULL)
```

**Arguments**

- `date`: A POSIXct vector containing the date(time) of the events.
- `skip`: A data.frame containing two POSIXct columns `start` and `end` describing the periods to skip over.
- `noskip`: A data.frame like `skip` but where the periods define the NON skipped part of the events.

**Details**

Interevents are the time differences between successive dates. When the `date` argument contains occurrence times $T_i$ for successive events of an homogeneous Poisson process, interevents $T_i - T_{i-1}$ are mutually independent with the same exponential distribution.

When some time intervals are skipped independently from the event point process, we may consider the interevents $T_i - T_{i-1}$ between two non-skipped events such that the time interval $(T_{i-1}, T_i)$ does not contain any skipped interval. These interevents still are mutually independent with the same exponential distribution. When `skip` or `noskip` is not `NULL` the computation therefore only retains couples of two successive datetimes "falling" in the same non-skipped period, which number can therefore be associated with the interevent.

**Value**

A list mainly containing a `interevt` data.frame.

- `interevt`: Data.frame. Each row describes a retained interevent through a period integer giving the "noskip" period, a start and end POSIX and a duration in days.
- `noskip`: Only hen `skip` or `noskip` args have been given. A data.frame containing broadly the same information as the `noskip` arg is it was given or the information deduced from the `skip` arg if given.
- `axis`: When needed, a list with some material to build an axis with uneven ticks as in the `gof.date` with `skip.action = "omit"`
Note

Only one of the two arguments `skip` and `noskip` should be given in the call. In each case, the rows of the returned data.frame objects describe periods in chronological order. That is: start at row 2 must be after the end value of row 1 and so on.

Note that there are usually less interevents than dates since two successive dates will be retained for an interevent only when they are not separated by missing period. As a limit case, there can be no interevents if the `noskip` periods contain only one date from the `date` vector.

Author(s)

Yves Deville

See Also

gof.date for goodness-of-fit diagnostics for dates of events expplot for diagnostics concerning the exponential distribution.

Examples

data(Brest)
ie <- interevt(date = Brest$OTdata$date, skip = Brest$OTmissing)

expplot(ie$interevt$duration, rate = 1 / mean(ie$interevt$duration),
        main = "No threshold")

## keep only data over a threshold
ind1 <- Brest$OTdata$Surge >= 35
ie1 <- interevt(Brest$OTdata$date[ind1], skip = Brest$OTmissing)
expplot(ie1$interevt$duration, main = "Threshold = 35")

## increase threshold
ind2 <- Brest$OTdata$Surge >= 55
ie2 <- interevt(date = Brest$OTdata$date[ind2], skip = Brest$OTmissing)
expplot(ie2$interevt$duration, main = "Threshold = 55 cm")

Lomax

Lomax distribution

Description

Density function, distribution function, quantile function and random generation for the Lomax distribution.
Usage

dlomax(x, scale = 1.0, shape = 4.0, log = FALSE)
plomax(q, scale = 1.0, shape = 4.0, lower.tail = FALSE)
qlomax(p, scale = 1.0, shape = 4.0)
rlomax(n, scale = 1.0, shape = 4.0)

Arguments

x, q        Vector of quantiles.
p           Vector of probabilities.
n           Number of observations.
scale, shape   Shift and shape parameters. Vectors of length > 1 are not accepted.
log           Logical; if TRUE, the log density is returned.
lower.tail Logical; if TRUE (default), probabilities are Pr[X <= x], otherwise, Pr[X > x].

Details

The lomax distribution function with shape \( \rho > 0 \) and scale \( \delta > 0 \) has distribution function

\[
F(y) = 1 - \left( \frac{\delta}{\delta + y} \right)^\rho \quad y > 0
\]

This distribution has increasing hazard and decreasing mean residual life (MRL). The coefficient of variation decreases with \( \rho \) and tends to 1 for large \( \rho \). The default value \( \rho = 4 \) corresponds to \( CV = \sqrt{2} \).

Value

dlomax gives the density function, plomax gives the distribution function, qlomax gives the quantile function, and rlomax generates random deviates.

Note

This distribution is sometimes called log-exponential and also called the Lomax distribution. It is occasionally used in POT with scale taken as the threshold.

References


Examples

library(Renext)
shape <- 5; scale <- 10
xl <- qlomax(c(0.00, 0.99), scale = scale, shape = shape)
x <- seq(from = xl[1], to = xl[2], length.out = 200)
f <- dlomax(x, scale = scale, shape = shape)
plot(x, f, type = "l", main = "Lomax distribution density")
MixExp2

MixExp2

Mixture of two exponential distributions

Description

Probability functions associated to the mixture of two exponential distributions.

Usage

dmixexp2(x, prob1,
        rate1 = 1.0, rate2 = rate1 + delta, delta,
        log = FALSE)

pmixexp2(q, prob1,
        rate1 = 1.0, rate2 = rate1 + delta, delta,
        log = FALSE)

qmixexp2(p, prob1,
        rate1 = 1.0, rate2 = rate1 + delta)

rmixexp2(n, prob1,
        rate1 = 1.0, rate2 = rate1 + delta)

hmixexp2(x, prob1,
        rate1 = 1.0, rate2 = rate1 + delta)

Arguments

x, q
Vector of quantiles.

p
Vector of probabilities.

n
Number of observations.

log
Logical; if TRUE, the log density is returned.

prob1
Probability weight for the "number 1" exponential density.

rate1
Rate (inverse expectation) for the "number 1" exponential density.

rate2
Rate (inverse expectation) for the "number 2" exponential density. Should in most cases be > rate1. See Details.

delta
Alternative parametrisation delta = rate2 - rate1.
Details

The density function is the mixture of two exponential densities

\[ f(x) = \alpha_1 \lambda_1 e^{-\lambda_1 x} + (1 - \alpha_1) \lambda_2 e^{-\lambda_2 x} \quad x > 0 \]

where \( \alpha_1 \) is the probability given in \texttt{prob1} while \( \lambda_1 \) and \( \lambda_2 \) are the two rates given in \texttt{rate1} and \texttt{rate2}.

A 'naive' identifiability constraint is

\[ \lambda_1 < \lambda_2 \]

i.e. \texttt{rate1} < \texttt{rate2}, corresponding to the simple constraint \( \text{delta} > 0 \). The parameter \( \text{delta} \) can be given instead of \texttt{rate2}.

The mixture distribution has a decreasing hazard, increasing Mean Residual Life (MRL) and has a thicker tail than the usual exponential. However the hazard, MRL have a finite non zero limit and the distribution behaves as an exponential for large return levels/periods.

The quantile function is not available in closed form and is computed using a dedicated numerical method.

Value

dmiwexp2, pmiwexp2, qmiwexp2, evaluates the density, the distribution and the quantile functions.
dmixexp2 generates a vector of \( n \) random draws from the distribution. hmixexp2 gives hazard rate and Hmixexp2 gives cumulative hazard.

Examples

```r
rate1 <- 1.0
rate2 <- 4.0
prob1 <- 0.8
qs <- qmixexp2(p = c(0.99, 0.999), prob1 = prob1,
               rate1 = rate1, rate2 = rate2)
x <- seq(from = 0, to = qs[2], length.out = 200)
F <- pmixexp2(x, prob1 = prob1, rate1 = rate1, rate2 = rate2)
plot(x, F, type = "l")
abline(v = qs[1])
abline(h = 0.99)
```

mom.mixexp2  

\textit{Moment estimation for the mixture of two exponential distributions}

Description

Compute the moment estimation for the tree parameters of the mixture of two exponential distributions.

Usage

```r
mom.mixexp2(x)
```
mom.mixexp2

Arguments

x          Sample. Vector containing values > 0.

Details

The three parameters (probability and the two rates) are computed from the first three moments (theoretical and sample). It can be shown that the inverse rates are obtained solving a quadratic equation. However the roots can be negative or complex and the estimates are not valid ones.

Value

A list with elements

  estimate A vector with named elements "prob1", "rate1" and "rate2". When the moment estimators are not valid (negative or complex rates), a vector of three NA is returned.
  method   Character "moments"

Note

The theoretical coefficient of variation (CV) of a mixture of two exponential distributions always exceed 100%. When the sample CV is over 100%, no valid estimates exist since the two first moments can not be matched.

Author(s)

Yves Deville

References


See Also

See `ini.mixexp2` for a more versatile initial estimation.

Examples

```r
x <- rmixexp2(n = 100, prob1 = 0.5, rate1 = 1.0, rate2 = 3.0)
est <- mom.mixexp2(x)
```
Parameters from moments

Description

Compute parameters from (theoretical) moments

Usage

```r
mom2par(densfun = "exponential",
        mean,
        sd = NA)
```

Arguments

- `densfun` Name of the distribution. This can be at present time: "exponential", "weibull", "gpd", "gamma", "negative binomial".
- `mean` Theoretical mean (expectation) of the distribution. Can be a vector, in which case the parameters will be vectors.
- `sd` Standard deviation.

Details

For some distributions like Weibull, it is necessary to find a numerical solution since the parameters have no closed form expression involving the moments.

Value

A named list containing the parameters values e.g. with names shape and scale. When mean or sd is vector the list contains vectors.

Note

The name of the formal argument `densfun` is for compatibility with `fitdistr` from the MASS package. However, unlike in `fitdistr` this formal cannot be given a density value, i.e. an object of the class "function" such as `dnorm`.

Author(s)

Yves Deville

Examples

```r
## Weibull
mom2par(densfun = "weibull", mean = 1, sd = 2)
## Genrealised Pareto
mom2par(densfun = "gpd", mean = 1, sd = 2)
## Gamma
mom2par(densfun = "gamma", mean = 1:10, sd = 1)
```
The vector \( \mathbf{N} \) contains counts for events occurring on non-overlapping time periods with lengths given in \( \mathbf{w} \). Under the NB Lévy process assumptions, the observed counts (i.e. elements of \( \mathbf{N} \)) are independent random variables, each following a negative binomial distribution. The size parameter \( r_k \) for \( N_k \) is \( r_k = \gamma w_k \) and the probability parameter \( p \) is \( \text{prob} \). The vector \( \mu \) of the expected counts has elements

\[
\mu_k = E(N_k) = \frac{1 - p}{p} \gamma w_k
\]

The parameters \( \gamma \) and \( p \) (\( \text{prob} \)) are estimated by Maximum Likelihood using the likelihood concentrated with respect to the \( \text{prob} \) parameter.
Value

A list with the results

- `estimate` Parameter estimates.
- `sd` Standard deviation for the estimate.
- `score` Score vector at the estimated parameter vector.
- `info` Observed information matrix.
- `cov` Covariance matrix (approx.).

Note

The Negative Binomial Lévy process is an alternative to the homogeneous Poisson point process when counts are subject to overdispersion. In the NB process, all counts share the same index of dispersion (variance/expectation ratio), namely 1/prob. When prob is close to 1, the counts are nearly Poisson-distributed.

Author(s)

Yves Deville

References


See Also

`NegBinomial` for the negative binomial distribution, `glm.nb` from the MASS package for fitting Generalised Linear Model of the negative binomial family.

Examples

```r
## known parameters
nint <- 100
gam <- 6
prob <- 0.20

## draw random w, then the counts N
w <- rgamma(nint, shape = 3, scale = 1/5)
N <- rnbinom(nint, size = w*gam, prob = prob)
mu <- w*gam*(1-prob)/prob
Res <- NBlevy(N = N, w = w)

## Use example data 'Brest'
data(Brest)

gof1 <- gof.date(date = Brest$OTdata$date,
                 skip = Brest$OTmissing,
```
OTjitter

Add a small amount of noise to a numeric vector

Description

Add a small amount of noise to a numeric vector keeping all the values above the given threshold.

Usage

OTjitter(x, threshold = NULL)

Arguments

x The numeric vector to which jitter should be added.

threshold A threshold above which all elements of the modified vector must stay.

Value

A vector with the same length and nearly the same values as x. As in jitter, a small amount of noise is added to each value of x. The noise level is adjusted so that every noisy value remains above the specified threshold. When the a value is very close to the threshold, only a very small amount of negative noise can be added.
Note

The aim of this function is to remove possible ties in experimental OT data. Ties cause problems or warnings in some goodness-of-fit tests such as Kolmogorov-Smirnov.

Author(s)

Yves Deville

See Also

jitter

Examples

data(Garonne)
x <- Garonne$OTdata$Flow
min(x)
yleft <- OTjitter(x, threshold = 2500)
length(x)
nlevels(as.factor(x))
nlevels(as.factor(xmod))
max(abs(x-xmod))

plot.Rendata

Plot a Rendata object

Description

Plot 'Rendata' datasets with OT and historical data

Usage

## S3 method for class 'Rendata'
plot(x,
textOver = quantile(x$OTdata[, x$info$varName], probs = 0.99),
showHist = TRUE,
...)

Arguments

x Rendata object i.e. a list object as read with the readXML function.
textOver Mark values of the variable in the OTdata part of x. Values above the textOver value (if any) will be marked with the character version of the block, typically a year
showHist If True, the historical periods (is any) are shown on the plot.
... further args to be passed to plot function.
Details

The plot shows the main data of the object `x` (the OTdata part) as well as historical data MAXdata or OTSdata if any. Different colours are used on the background. This function is not intended to produce nice plots to be printed.

Note

This function is mainly a companion function of `readXML`. Its goal is to check the content of the data read.

Author(s)

Yves Deville

See Also

`readXML`

Examples

```r
library(XML)
## use 'index.xml' file shipped with Renext
dir1 <- system.file("Rendata", package = "Renext")
BrestNew <- readXML(name = "Brest", dir = dir1)
plot.Rendata(BrestNew)
GaronneNew <- readXML(name = "Garonne", dir = dir1)
plot.Rendata(GaronneNew)
test1 <- readXML(name = "test1", dir = dir1)
plot.Rendata(test1)
```

Description

Plot an object of class "Renouv". The plot is a return level plot with some supplementary elements to display historical data.

Usage

```r
## S3 method for class 'Renouv'
plot(x,
pct.conf = NULL,
show.MAX = TRUE, show.OTS = TRUE,
mono = TRUE,
rl.mark = NULL, labels.mark = rl.mark, col.mark = NULL,
main = NULL,
problim = NULL,
```
Tlim = NULL,
xlab = "periods",
ylab = "level",
...

Arguments

x Object of class "Renouv".
pct.conf Percents for confidence limits (lower and upper). These levels should be found within those computed in the object x. By default, all computed levels will be used.
show.MAX Logical. If TRUE and is x embeds historical data of type MAX, then these will be shown with a symbol differing from the one for ordinary points.
show.OTS Logical. If TRUE and is x embeds historical data of type OTS, then these will be shown with a symbol differing from the one for ordinary points. An exception is when one or several OTS block have no data. Then each such block is shown as an horizontal segment; its right end-point shows the effective duration of the block and the ordinate shows the OTS threshold for this block. No data exceeded the threshold within the block.
mono Logical. For a monochrome plot.
rl.mark Each return level is materialised with a vertical line.
labels.mark Label(s) for marked return levels.
col.mark Color(s) for marked return levels.
main Main title.
problim Limits for the x-axis in probability scale.
Tlim Limits for the x-axis in return period scale. The values are given as a numeric vector of length 2, containing values >=0. The first element (minimal return period can be 0 in which case it will be replaced by a very small positive value.
xlab Label of the x-axis (log scale).
ylab Label of the y-axis.
... Further arguments to be passed to plot e.g. ylim to adjust the y-axis.

Details

The return level plot is of exponential type i.e. uses a log-scale for return periods.

Examples

data(Brest)
fitB <- Renouv(Brest)
plot(fitB, main = "POT fit for Brest data", xlab = "years")
## Garonne example contains historical data.
data(Garonne)
fitG <- Renouv(Garonne)
plot(fitG, main = "POT fit for Garonne data", xlab = "years")
plot(fitG, Tlim = c(1, 2000))
`predict.Renouv`  

**Description**  
Compute return levels and confidence limits for an object of class "Renouv".

**Usage**  
```r  
## S3 method for class 'Renouv'  
predict(object,  
    newdata = c(10, 20, 50, 100, 200, 500, 1000),  
    cov.rate = FALSE,  
    level = c(0.95, 0.7),  
    trace = 1, eps = 1e-06,  
    ...)  
```

**Arguments**  
- `object`: An object of class "Renouv" typically created by using the `Renouv` function.  
- `newdata`: The return period at which return levels and confidence bounds are wanted.  
- `cov.rate`: If `FALSE`, the delta method will not take into account the uncertainty on the event rate `lambda` of the Poisson process. NOT IMPLEMENTED YET. At the time only `FALSE` is possible.  
- `level`: Confidence levels as in other `predict` methods (not percentages).  
- `trace`: Some details are printed when `trace` is not zero.  
- `eps`: Level of perturbation used to compute the numerical derivatives in the delta method.  
- `...`: Further arguments passed to or from other methods.

**Details**  
Unless in some very special cases, the confidence limits are approximated ones computed by using the delta method with numerical derivatives.

**Value**  
A `data.frame` with the expected return levels (col. named "quant") at the given return periods, and confidence limits. The returned object has an `infer.method` attribute describing the method used to compute the confidence limits.
Note

Despite of its name, this method does not compute true predictions. A return period is to be interpreted as an average interevent time rather than the duration of a specific period of time. For instance, the expected return level for a given return period with length 100 years is the level that would be on average exceeded once every 100 years (assuming that the model description in object is correct).

Author(s)

Yves Deville

References


See Also

Renouv to fit Renouv model.

Examples

```r
library(Renext)
data(Brest)
fit <- Renouv(Brest)
pred <- predict(fit, newdata = c(100, 125, 150, 175, 200),
                 level = c(0.99, 0.95))
```

---

**readXML**

*Read data using an XML index file*

**Description**

Read one or several dataset(s) using an XML index file specifying the data sets to read and the structure of each

**Usage**

```r
readXML(name,
       dir,
       index = "index.xml",
       trace = 0)
```
readXML

Arguments

- **name**: Name for the dataset that will be matched against the name attribute of datasets as they are given in the index file.
- **dir**: Path to the directory where the index file and all data files should be found.
- **index**: Name (short) of the index file. This file must be in the directory given by `dir`.
- **trace**: Integer or logical to trace the successive steps by short indications.

Details

The XML index file is parsed within R. Then according to the indications within the index file, other files are read (e.g. csv files). In this way, data returned as a list can contain heterogeneous data: Over Threshold (OT) data, missing periods, MAX data, etc. Various pieces of information are also stored in list elements with name containing the "info" string.

This function requires the CRAN package XML.

Value

A list with the data read.

- **info**: General information about the data: `varName`, `varShortLabel` and `varUnit` give the variable name unit and short label.
- **OTinfo**: Information for the Over the Threshold (OT).
- **OTdata**: Over the Threshold (OT) data.
- **OTmissing**: Missing periods within the OTdata period.
- **MAXinfo**: Information for the MAX (r-largest) supplement data.
- **MAXdata**: MAX supplement data.
- **OTSinfo**: Information for the Over the Threshold Supplement (OTS) data.
- **OTSdata**: Over the Threshold (OT) supplement data.

Note

The flat files (usually .csv files) can also be read in a more conventional way e.g. through `read.table`. However, conform to the `index.xml` examples or to the `index.xsd` schema to see how to adjust the reading of parameters such as `sep`, etc.

Author(s)

Yves Deville

See Also

See Brest for an example of such a list.

See URL [https://gforge.irsn.fr/gf/project/renext](https://gforge.irsn.fr/gf/project/renext) for data related to Renext.
Examples

```r
## Not run:
library(XML)
## use 'index.xml' file shiped with Renext
dir1 <- system.file("Rendata", package = "Renext")
BrestNew1 <- readXML(name = "Brest", dir = dir1)
test1 <- readXML(name = "test1", dir = dir1)

## End(Not run)
```

---

**Renouv**

*Fit a 'Renouvellement' model*

**Description**

Fit a 'renouvellement' POT model using Over the Threshold data and possibly historical data of two kinds.

**Usage**

```r
Renouv(x, 
  threshold = NULL, 
  effDuration = NULL, 
  distname.y = "exponential", 
  MAX.data = NULL, 
  MAX.effDuration = NULL, 
  OTS.data = NULL, 
  OTS.effDuration = NULL, 
  OTS.threshold = NULL, 
  fixed.par.y = NULL, 
  start.par.y = NULL, 
  force.start.H = FALSE, 
  numDeriv = TRUE, 
  trans.y = NULL, 
  jitter.KS = TRUE, 
  pct.conf = c(95, 70), 
  rl.prob = NULL, 
  prob.max = 1.0-1e-04, 
  pred.period = NULL, 
  suspend.warnings = TRUE, 
  control = list(maxit = 300, fnscale = -1), 
  control.H = list(maxit = 300, fnscale = -1), 
  trace = 0, 
  plot = TRUE, 
  ...)
```
Arguments

x Either a numeric vector or an object of the class "Rendata". In the first case, x contains all the levels above the threshold for a variable of interest. In the second case, most formal arguments take values in accordance with the object content, and can be by-passed by giving the formal explicitly.

threshold Value of the threshold for the OT data.

effDuration Effective duration, i.e. duration of the OT period.

distname.y Name of the distribution for the exceedances over the threshold.

MAX.data Either numeric vector or a list of numeric vectors representing historical data $r$-max by blocks. When a vector is given, there is only one block, and the data are the corresponding $r$-max observed levels where $r$ is the vector length; the block duration is given in OTS.effDuration. When a list is given, each list element contains the data for one block, and the effective duration are in OTS.effDuration

MAX.effDuration Vector of (effective) durations, one by block MAX data.

OTS.data A numeric vector or a list of numeric vectors representing supplementary Over Threshold data in blocks. When a vector is given, there is only one block, and the data contain all the 'historical' levels over the corresponding threshold given in OTS.threshold. The block duration is given in OTS.effDuration. When a list is given, each list element contains the data for one block, and the threshold and effective duration are in OTS.threshold and OTS.effDuration.

OTS.effDuration A numeric vector giving the (effective) durations for the OTS blocks.

OTS.threshold A vector giving the thresholds for the different OTS blocks.

fixed.par.y Named list of known (or fixed) parameter values for the $y$-distribution.

start.par.y Named list of parameter initial values for the $y$-distribution. Only used when the distribution does not belong to the list of special distributions.

force.start.H Logical. When TRUE, the values in start.par.y (which must then be correct) will be used also as starting values in the maximisation of the global likelihood: OT data and historical data. This is useful e.g. when the historical data fall outside of the support for the distribution fitted without historical data. See below the Details section.

numDeriv Logical: should the hessian be computed using the numDeriv package (value TRUE) or should it be taken from the results of optim?

trans.y Transformation of the levels before thresholding (if not NULL). This is only possible with the "exponential" value distname.y. The two allowed choices are "square" and "log" meaning that the fitted (exponentially distributed) values are $x.OT^2 - \text{threshold}^2$ and $\log(x.OT) - \log(\text{threshold})$ respectively. The corresponding distributions for $x.OT$ may be called "square-exponential" and "log-exponential".

jitter.KS Logical. When set to TRUE, a small amount of noise is added to the "OT" data used in the Kolmogorov-Smirnov test in order to remove ties. This is done using the OTjitter function.
pct.conf  Character or numeric vector specifying the percentages for the confidence (bi-
lateral) limits on quantiles.

r1.prob  Vector of probabilities for the computation of return levels. These are used
in plots (hence must be dense enough) and appear on output in the data.frame
ret.lev.

prob.max  Max value of probability for return level and confidence limits evaluations. This
argument 'shortens' the default prob vector: values > prob.max in the default
prob vector are omitted. Ignored when a prob argument is given.

pred.period  A vector of "pretty" periods at which return level and probability will be evalu-
ated and returned in the pred data.frame.

suspend.warnings  Logical. If TRUE, the warnings will be suspended during optimisation steps. This
is useful when the parameters are subject to constraints as is usually the case.

control  A named list used in optim for the no-history stage (if any). Note that fnscale = -1
says that maximisation is required (not minimisation) and must not be changed!

control.H  A named list used in optim for the historical stage (if any).

trace  Level of verbosity. Value 0 prints nothing.

plot  Draw a return level plot?

...  Arguments passed to plot.Renouv, e.g. main, ylim.

Details

The model is fitted using Maximum Likelihood (ML).

Some distributions listed below and here called "special" are considered in a special manner. For
these distributions, it is not necessary to give starting values nor parameter names which are unam-
biguous.

<table>
<thead>
<tr>
<th>distribution</th>
<th>parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>exponential</td>
<td>rate</td>
</tr>
<tr>
<td>weibull</td>
<td>shape, scale</td>
</tr>
<tr>
<td>gpd</td>
<td>scale, shape</td>
</tr>
<tr>
<td>log-normal</td>
<td>meanlog, sdlog</td>
</tr>
<tr>
<td>gamma</td>
<td>shape, scale</td>
</tr>
<tr>
<td>mixexp2</td>
<td>prob1, rate1, delta</td>
</tr>
</tbody>
</table>

Other distributions can be used. Because the probability functions are then used in a "black-box"
fashion, these distributions should respect the following formal requirements:

1. The name for the density, distribution and quantile functions must obey to the classical "pre-
fixing convention". Prefixes must be respectively: "d", "p", "q". This rules applies for distrib-
ution of the stats package and those of many other packages such evd.

2. The first (main) argument must be vectorisable in all three functions, i.e. a vector of x, q or p
must be accepted by the density, the distribution and the quantile functions.

3. The density must have a log formal argument. When log is TRUE, the log-density is returned
instead of the density.
For such a distribution, it is necessary to give arguments names in start.par.y. The arguments list must have exactly the required number of parameters for the family (e.g. 2 for gamma). Some parameters can be fixed (known); then the parameter set will be the reunion of those appearing in start.par.y and those in fixed.par.y. Anyway, in the present version, at least one parameter must be unknown for the y part of model.

Mathematical requirements exist for a correct use of ML. They are referred to as "regularity conditions" in ML theory. Note that the support of the distribution must be the set of non-negative real numbers.

The estimation procedure differs according to the existence of historical (MAX) data.

1. When no historical data is given, the whole set of parameters contains orthogonal subsets: a "point process" part concerning the process of events, and an "observation" part concerning the threshold exceedances part. The parameters can in this case be estimated separately. The rate of the Poisson process is estimated by the empirical rate, i.e. the number of events divided by the total duration given in effDuration. The "Over the Threshold" parameters are estimated from the exceedances computed as x.OT minus the threshold.

2. When historical data is given, the two parameter vectors must be coped with together in maximising the global likelihood. In this case, we begin the estimation ignoring the historical data and then use the estimates as starting values for the maximisation of the global likelihood. In some circumstances, the estimates obtained in the first stage can not be used with historical data because some of these fall outside the support of the distribution fitted. This can happen e.g. with a distname.y = "gpd" when historical data exceed threshold - scale/shape for the values of shape and scale computed in the first stage.

Value

An object with class "Renouv". This is mainly a list with the various results.

- **est.N**: Estimate(s) for the count "N" part. This estimate does not use the historical data, even if available.
- **est.y**: Estimate(s) for the exceedance "y" part. This estimate does not use the historical data, even if available.
- **cov.N**, **cov.y**: The (co-)variances for the estimates above.
- **estimate**: Estimate(s) for the whole set of parameters based on OT data and on historical data if available.
- **ks.test**: Kolmogorov-Smirnov goodness-of-fit test.
- **ret.lev**: A data.frame containing return levels and confidence limits. The corresponding probabilities are either provided by user or taken as default values.
- **pred**: A data.frame similar to ret.lev, but with "pretty" return periods. These are taken as the provided values pred.period if any or are chosen as "round" multiples of the time unit (taken from effDuration). The periods are chosen in order to cover periods ranging from 1/10 to 10 time units.

Other results are available. Use names(result) to see their list.

Except in the the special case where distname.y is "exponential" and where no historical data are used, the inference on quantiles is obtained with the delta method and using numerical derivatives. Confidence limits are unreliable for return levels much greater than the observation-historical period.
Due to the presence of estimated parameters, the Kolmogorov-Smirnov test is unreliable when less than 30 observations are available.

**Warning**

With some distributions or in presence of historical data, the estimation can fail due to some problem during the optimisation. Even when the optimisation converges, the determination of the (numerical) hessian can be impossible: This can happen if *one or more parameter is too small* to compute a finite difference approximation of gradient. For instance the ’rate’ parameter of the exponential distribution (= inverse mean) will be small when the mean of exceedances is large.

A possible solution is then to **rescale the data** e.g. dividing them by 10 or 100. As a rule of thumb, an acceptable scaling leads to data (exceedances) of a few units to a few hundreds, but **an order of magnitude of thousands or more should be avoided and reduced by scaling**. The rescaling is recommended for the square exponential distribution (obtained with `trans = "square"`) since the observations are squared.

Another possible way to solve the problem is to change the `numDeriv` value.

This problem will be solved in a future version.

**Note**

The model only concerns the "Over the Threshold" part of the distribution of the observations. When historical data is used, observations should all be larger than the threshold.

The name of the elements in the returned list is indicative, and is likely to be changed in future versions. At the time, the effect of historical data on estimation (when such data exist) can be evaluated by comparing \( c(res\$est.N, res\$est.y) \) and \( res\$estimate \) where \( res \) is the results list.

Some warnings may indicate that missing values are met during the optimisation process. This is due to the evaluation of the density at tail values. At the time the ML estimates are computed using an unconstrained optimisation, so invalid parameter values can be met during the maximisation or even be returned as (invalid) estimates.

Validity tests for the estimation in presence of historical data have been limited at the time. Therefore this possibility should be regarded as experimental.

**Author(s)**

Yves Deville

**References**


**See Also**

`rRenouv` to simulate "renouvellement" data, `RLplot` for the return level plot. See `optim` for the tuning of optimisation.
Examples

```r
library(Renext)
data(Garonne)

## use a "Rendata" object as 'x'. Historical data are used!
fit <- Renouv(x = Garonne, distname = "weibull", trace = 1,
        main = "'Garonne' data")
summary(fit)

## generates a warning because of the ties
fit2 <- Renouv(x = Garonne, distname = "gpd",
        jitter.KS = FALSE,
        threshold = 2800, trace = 1,
        main = "'Garonne' data with threshold = 2800 and GPD fit")

## use a numeric vector as 'x'
fit3 <- Renouv(x = Garonne$OTdata$Flow,
        threshold = 2500,
        effDuration = 100,
        distname = "gpd",
        OTS.data = list(numeric(), c(6800, 7200)),
        OTS.effDuration = c(100, 150),
        OTS.threshold = c(7000, 6000),
        trace = 1,
        main = "'Garonne' data with artificial "OTS" data")

## Add historical (fictive) data
fit4 <- Renouv(x = Garonne$OTdata$Flow,
        threshold = 2500,
        effDuration = 100,
        distname = "weibull",
        fixed.par.y = list(shape = 1.1),
        OTS.data = list(numeric(), c(6800, 7200)),
        OTS.effDuration = c(100, 150),
        OTS.threshold = c(7000, 6000),
        trace = 0,
        main = "'Garonne' data with artificial "OTS" data")
```

Description

Return level plot for "Renouvellement" models
Usage

RLplot(data,
       x = NULL,
       duration = 1,
       lambda,
       conf.pct = 95,
       mono = TRUE,
       mark.rl = 100,
       mark.labels = mark.rl,
       mark.col = NULL,
       main = NULL,
       ylim = NULL,
       ...)

Arguments

data A data.frame object with a column named quant
x Optional vector of observed levels
duration The (effective) duration corresponding to x if this argument is used.
lambda Rate (in accordance with duration.
conf.pct a vector (character or integer) giving confidence levels. See Details below.
mono If TRUE colours are replaced by black.
mark.rl Return levels to be marked on the plot.
mark.labels labels shown at positions in mark.rl.
mark.col Colours for marked levels.
main Main title for the return level plot (defaults to empty title).
ylim Limits for the y axis (defaults to values computed from the data).
... Further args to be passed to plot. Should be removed in future versions.

Details

Per cents should match column names in the data.frame as follows. The upper and lower limits are expected to be U.95 and L.95 respectively. For a 70% confidence percentage, columns should have names "U.70" and "L.70".

The plot is comparable to the return level described in Coles' book and related packages, but the return level is here in log-scale while Coles uses a loglog-scale. A line corresponds here to a one parameter exponential distribution, while Coles' plot corresponds to Gumbel. Since \( \log(-\log(1-p)) \) is close to \( \log(p) \) when \( p \) is close to 1, the two plots differ only for small return levels. This plot is identical to an expplot but with x and y scales changed. Only axis tick-marks differ. The convexity of the scatter plot is therefore opposed in the two plots.
Note

Confidence limits correspond to two-sided symmetrical intervals. This means that the (random) confidence interval may be under or above the true unknown value with the same probabilities. E.g. the probability that the unknown quantile falls above \( U_{.95} \) is 2.5%. The two bounds are yet generally not symmetrical with respect to quant; such a behaviour follows from the use of "delta" method for approximate intervals.

It is possible to add graphical material (points, lines) to this plot using \( \log(\text{returnlev}) \) and quantile coordinates. See Examples section.

Author(s)

Y. Deville

References


See Also

See expplot for a classical exponential plot. See Also as Renouv to fit "Renouvellement" models. The return.level function in the extRemes package.

Examples

```r
## Typical probability vector
prob <- c(0.0001,
   seq(from = 0.01, to = 0.09, by = 0.01),
   seq(from = 0.10, to = 0.80, by = 0.10),
   seq(from = 0.85, to = 0.99, by = 0.01),
   0.995, 0.996, 0.997, 0.998, 0.999, 0.9995)

## Model parameters rate = #evts by year, over nyear
lambda <- 4
nyear <- 3
theta.x <- 4

## draw points
n.x <- rpois(1, lambda = lambda*nyear)
x <- rexp(n.x, rate = 1/theta.x)

## ML estimation (exponential)
lambda.hat <- n.x / nyear
theta.x.hat <- mean(x)

## Compute bounds (here exact)
alpha <- 0.05
quant <- qexp(p = prob, rate = 1/theta.x.hat)

theta.L <- 2*n.x*theta.x.hat / qchisq(1 - alpha/2, df = 2*n.x)
theta.U <- 2*n.x*theta.x.hat / qchisq(alpha/2, df = 2*n.x)
```
L.95 <- qexp(p = prob, rate = 1/theta.L)
U.95 <- qexp(p = prob, rate = 1/theta.U)

## store in data.frame object
data <- data.frame(prob = prob, quant = quant, L.95 = L.95, U.95 = U.95)

RLplot(data = data, x = x, lambda = lambda.hat,
       duration = nyear,
       main = "Poisson-exponential return levels")

RLplot(data = data, x = x, lambda = lambda.hat, duration = nyear,
       mark.r1 = 10, mark.labels = "10 ans", mono = FALSE, mark.col = "SeaGreen",
       main = "Poisson-exponential return levels")

points(x = log(5/zero.noslash), y = 25, pch = 18, cex = 1.4, col = "purple")
text(x = log(5/zero.noslash), y = 25, col = "purple", pos = 4, labels = "special event")

---

roundPred

Round quantiles in a pseudo-prediction table.

Description

Round the quantiles of a pseudo prediction table such that computed by predict.Renouv.

Usage

roundPred(pred, dig.quant = NA)

Arguments

pred The data.frame containing the predicted quantiles and return levels.
dig.quant Number of digits. Guessed if not provided.

Details

Only the columns that can be considered as quantiles are rounded. These are assumed to have names "quant" for the expected return level and "L." or "U." followed by a percentage for lower and upper confidence limits (e.g. "L.95" and "U.95" for 95% percent confidence limits. The number of digits guessed is experimental.

Value

A data.frame with the same structure as that given, but with some columns rounded.
Description

Simulation of renewal ("Renouvellement") data, i.e. observations Over a Threshold and their counts on blocks (e.g. years).

Usage

```r
rRenouv(densfun.y = "exponential",
        par.y = list(rate = 1),
        densfun.N = "poisson",
        par.N = list(lambda = 6),
        threshold = 0,
        aggreg = TRUE,
        nb = 50,
        labb = seq(to = 2009, by = 1, length = nb),
        w = rep(1, nb))
```

Arguments

densfun.y: A character string specifying the distribution of the exceedances over the threshold. At the time only "exponential", "weibull" and "gpd" are available.

par.y: Named list giving parameter values for the "y"-distribution. See examples below.

densfun.N: Character string specifying the distribution for the counts. At the time only "poisson" and "negative binomial" are allowed.

par.N: Named list giving parameter values for the N-distribution for counts. The names are lambda in the Poisson case ("poisson"), and size and prob in the negative binomial case ("negative binomial" case).

threshold: Threshold value.

aggreg: Only TRUE is possible at the time.

nb: Number of blocks (or time intervals).

labb: Numeric vector of length nb that will be used in replacement of the block numbers vector. Typically it can contain year numbers. Use NULL to obtain blocks from 1 to nb.

w: Vector of blocks (time interval) length, i.e. duration.

Value

A list with the following objects

x: Vector of x values i.e. threshold plus exceedances.

N: Vector of counts.
block Vector of length \texttt{length(x)} giving the block number for the corresponding element in \texttt{x}. When coerced to a factor \texttt{block} has \texttt{length(N)} levels.

Note

At the present time, the random drawings of the \texttt{gpd} distribution are generated with the the \texttt{rgpd} of the \texttt{evd} package.

In future versions, this function might return an object with a special class (with name such as "renouv"). Then classical \texttt{simulate}, \texttt{plot}, ... methods could be defined for that class.

Author(s)

Yves Deville

See Also

\texttt{Renouv} to fit Renouvellement models.

Examples

```r
## Not run:
require(evd)
test1 <- rRenouv(nb = 100,
    threshold = 40,
    par.N = list(lambda = 2),
    densfun.y = "gpd",
    par.y = mom2par("gpd", mean = 30, sd = 36))

## End(Not run)
test2 <- rRenouv(nb = 100,
    threshold = 40,
    par.N = list(lambda = 2),
    densfun.y = "weibull",
    par.y = mom2par("weibull", mean = 30, sd = 36))

test3 <- rRenouv(nb = 100,
    threshold = 40,
    densfun.N = "negative binomial",
    par.N = list(gamma = 10, prob = 0.7),
    densfun.y = "weibull",
    par.y = mom2par("weibull", mean = 30, sd = 36))
```
**skip2noskip**

Fix non-skipped periods from skipped ones

**Description**

Compute non-skipped periods form start and end of skipped periods.

**Usage**

```r
skip2noskip(skip = NULL,
             start = NULL,
             end = NULL)
```

**Arguments**

- `skip`: A data.frame object with `start` and `end` columns that can be coerced to POSIXct. Other columns can be present (and will be ignored). Each row describes a missing period. Rows must be sorted in chronological order and periods should not overlap. Validity checks are at the time very limited.
- `start`: Beginning of the whole period, to be used in `as.POSIXct`.
- `end`: End of the whole period to be used in `as.POSIXct`.

**Details**

In a 'normal' use of this function `start` and `end` are given, and are respectively before the beginning of the first skip period and after the end of the last skip period. Thus the returned dataframe will have `nrow(skip)+1` rows. However, `start` and `end` can be `NULL` in which case only the `nrows(skip)-1` "inner" non-skipped periods will be returned. If `start` and `end` are `NULL` and `skip` has only one row, the returned result is `NULL`.

**Value**

A data.frame object with two POSIXct columns named `start` and `end`. Each row corresponds to a non-skipped period

**Author(s)**

Yves Deville

**See Also**

`readXML` for reading data from XML and csv files.
Examples

```r
data(Brest)

ns <- skip2noskip(skip = Brest$OTmissing)
ns2 <- skip2noskip(skip = Brest$OTmissing, 
                   start = Brest$OTinfo$start, 
                   end = Brest$OTinfo$end)

## check durations. dur2 should be equal to the effective 
## duration (with an error of a fraction of day)
dur <- as.numeric(sum(ns$end-ns$start))/365.25
dur2 <- as.numeric(sum(ns2$end-ns2$start))/365.25
```

---

**SLTW**  
*Shifted Left Truncated Weibull (SLTW) distribution*

**Description**

Density function, distribution function, quantile function and random generation for the Shifted Left Truncated Weibull distribution.

**Usage**

```r
dSLTW(x, delta = 1./zero.noslash, shape = 1./zero.noslash, scale = 1./zero.noslash, log = FALSE)
pSLTW(q, delta = 1./zero.noslash, shape = 1./zero.noslash, scale = 1./zero.noslash, lower.tail = FALSE)
qSLTW(p, delta = 1./zero.noslash, shape = 1./zero.noslash, scale = 1./zero.noslash)
rSLTW(n, delta = 1./zero.noslash, shape = 1./zero.noslash, scale = 1./zero.noslash)
```

**Arguments**

- `x, q` Vector of quantiles.
- `p` Vector of probabilities.
- `n` Number of observations.
- `delta, shape, scale` Shift, shape and scale parameters. Vectors of length > 1 are not accepted.
- `log` Logical; if TRUE, the log density is returned.
- `lower.tail` Logical; if TRUE (default), probabilities are $\Pr[X \leq x]$, otherwise, $\Pr[X > x]$. 
Details

The SLTW distribution function with shape $\alpha > 0$, scale $\beta > 0$ and shift $\delta > 0$ has distribution function

$$F(y) = 1 - \exp \left\{ - \left( \frac{y + \delta}{\beta} \right)^{\alpha} - \left( \frac{\delta}{\beta} \right)^{\alpha} \right\} \quad y > 0$$

This distribution is that of $Y = X - \delta$ conditional to $X > \delta$ where $X$ follows a Weibull distribution with shape $\alpha$ and scale $\beta$.

The hazard and mean residual life (MRL) are monotonous functions with the same monotonicity as their Weibull equivalent (with the same shape and scale). The moments or even expectation do not have simple expression.

This distribution is sometimes called power exponential. It is occasionally used in POT with the shift $\delta$ taken as the threshold as it should be when the distribution for the level $X$ (and not for the exceedance $Y$) is known to be the standard Weibull distribution.

Value

dSLTW gives the density function, pSLTW gives the distribution function, qSLTW gives the quantile function, and rSLTW generates random deviates.

See Also

Lomax for the Lomax distribution which is a limit case of SLTW.

Examples

```r
shape <- rexp(1)+1
delta = 10
xl <- qSLTW(c(0.001, 0.99), delta = delta, shape = shape)
x <- seq(from = xl[1], to = xl[2], length.out = 200)
f <- dSLTW(x, delta = delta, shape = shape)
plot(x, f, type = "l", main = "SLTW density")
F <- pSLTW(x, delta = delta, shape = shape)
plot(x, F, type = "l", main = "SLTW distribution")
X <- rSLTW(5000, delta = delta, shape = shape)
# Should be close to uniform repartition
plot(ecdf(pSLTW(X, delta = delta, shape = shape)))
```

summary.Rendata

Summary method for "Rendata" objects

Description

Summary method for "Rendata" objects representing data to be used in renouvellement models.
## Usage

```r
## S3 method for class 'Rendata'
summary(object, ...)

## S3 method for class 'summary.Rendata'
print(x, ...)
```

### Arguments

- `object`: An object with class "Rendata".
- `x`: An object of class "summary.Rendata", i.e. a result of a call to `summary.Rendata`.
- `...`: Further arguments passed to or from other methods.

### Author(s)

Yves Deville

### Examples

```r
## no historical data
data(Brest)
summary(Brest)

## example with historical data
data(Garonne)
summary(Garonne)
```

## Description

Summary method for "Renouv" objects representing 'Renouvellement' (POT) fitted models.

## Usage

```r
## S3 method for class 'Renouv'
summary(object,
         correlation = FALSE,
         symbolic.cor = FALSE,
         ...
)

## S3 method for class 'summary.Renouv'
print(x,
      coef = TRUE,
      pred = TRUE,
      ...
)
```
### Arguments

- **object**: An object with class "Renouv".
- **x**: An object of class "summary.Renouv", i.e. a result of a call to `summary.Renouv`.
- **correlation**: Logical; if TRUE, the correlation matrix of the estimated parameters is returned and printed.
- **coef**: Logical. If FALSE, the table of coefficients and t-ratios' will not be printed.
- **pred**: Logical. If FALSE, the table of return periods/levels will not be printed.
- **probT**: If FALSE, the p-values for the t-tests will not be printed nor displayed.
- **digits**: The number of significant digits to use when printing.
- **symbolic.cor**: Logical. If TRUE, print the correlations in a symbolic form (see `symnum`) rather than as numbers.
- **signif.stars**: Logical. If TRUE, 'significance stars' are printed for each coefficient.
- **...**: Further arguments passed to or from other methods.

### Details

`print.summary.Renouv` tries to be smart about formatting the coefficients, standard errors, return levels, etc. `format.summary.Renouv` returns as a limited content as a character string. It does not embed coefficients values nor predictions.

### Value

The function `summary.Rendata` computes and returns a list of summary statistics concerning the object of class "Rendata" given in object. The returned list is an object with class "summary.Renouv".

The function `print.summary.Rendata` does not returns anything.

### See Also

The model fitting function `Renouv` (to build "Renouv" model objects), `summary`.

### Examples

```R
library(Renext)
data(Brest)
fit <- Renouv(Brest)
summary(fit)
```
**weibplot**

*Classical Weibull distribution plot*

**Description**

Plots a vector using Weibull distribution scales

**Usage**

```r
weibplot(x,
plot.pos = "exp",
shape = NULL, scale = NULL,
labels = NULL,
mono = TRUE,
...)
```

**Arguments**

- `x`: The vector to be plotted.
- `plot.pos`: Plotting position for points: either "exp" for expected ranks or "med" for a median rank approximation (see Details below).
- `shape`: Shape parameter for one or several Weibull lines to be plotted.
- `scale`: Scale parameter for one or several Weibull lines to be plotted.
- `labels`: Text to display in legend when Weibull lines are specified.
- `mono`: Monochrome graph.
- `...`: Arguments to be passed to `plot`.

**Details**

This plot shows \( \log\{-\log[1 - F(x)]\} \) against \( \log(x) \) where \( F(x) \) at point \( i \) is taken as \( i/(n + 1) \) if `plot.pos` is "exp", or as the "median rank" approximation \( (i - 0.3)/(n + 0.4) \) if `plot.pos` is "med".

**Note**

The graph displayed uses a log scale for `x`. The log-log scale for `y` is emulated via the construction of suitable graduations. So be careful when adding graphical material (points, etc) to this graph with functions of the "add to plot" family (`points`, `lines`, ...).

**Author(s)**

Yves Deville

**See Also**

The `expplot` function for an "exponential distribution" plot (dedicated to the `shape = 1` case), and the `fweibull` function for ML estimation of the parameters.
Examples

```r
x <- rweibull(200, shape = 1.2, scale = 1)
weibplot(x, main = "Classical Weibull plot")
## Weibull lines
weibplot(x, shape = c(0.9, 1.3), scale = 1)
weibplot(x, shape = c(0.9, 1.3), scale = 1,
         labels = c("before", "after"))
weibplot(x, shape = c(0.9, 1.3), scale = 1,
         labels = c("before", "after"),
         mono = TRUE)
```
Index

*Topic datagen
  Renext-package, 2

*Topic datasets
  Brest, 7
  Brest.years, 8
  Brest.years.missing, 9
  Dunkerque, 10
  Garonne, 15

*Topic distribution
  Lomax, 23
  MixExp2, 21, 25
  mom.mixexp2, 12, 21, 26
  mom2par, 28
  NBlevy, 29
  NegBinomial, 30
  optim, 40, 42
  OTjitter, 31, 32

barplotRenouv, 4
Brest, 7, 9, 11, 37
Brest.years, 8, 9
Brest.years.missing, 9, 9

dlomax (Lomax), 23
dmixexp2 (MixExp2), 25
dSLTW (SLTW), 50
Dunkerque, 10

EM.mixexp, 11
expplot, 12, 20, 23, 45, 54

format.summary.Renouv (summary.Renouv), 52
fweibull, 14, 54

Garonne, 11, 15
glm.nb, 30
gof.date, 17, 23
gofExp.test, 19

Hmixexp2 (MixExp2), 25
hmixexp2 (MixExp2), 25

ini.mixexp2, 12, 20, 27
interevt, 13, 18, 22

jitter, 31, 32
ks.test, 20
Lomax, 23, 51
MixExp2, 21, 25
mom.mixexp2, 12, 21, 26
mom2par, 28
NBlevy, 29
NegBinomial, 30
optim, 40, 42
OTjitter, 31, 32

plomax (Lomax), 23
plot.Rendata, 6, 32
plot.Renouv, 33, 40
pmixexp2 (MixExp2), 25
predict.Renouv, 35
print.summary.Rendata
  (summary.Rendata), 51
print.summary.Renouv (summary.Renouv), 52
pSLTW (SLTW), 50

qllomax (Lomax), 23
qmixexp2 (MixExp2), 25
qSLTW (SLTW), 50

readXML, 33, 36, 49
Renext (Renext-package), 2
Renext-package, 2
Renouv, 3, 36, 38, 45, 48, 53
rlomax (Lomax), 23
RLplot, 42, 43
rnmixexp2 (MixExp2), 25
roundPred, 46
rRenouv, 42, 47
rSLTW (SLTW), 50
INDEX

skip2noskip, 49
SLTW, 50
summary, 53
summary.Rendata, 51
summary.Renouv, 52
symnum, 53

weibplot, 13, 15, 54