Package ‘ismev’

July 18, 2009

Version 1.34

Date 2009-14-07

Title An Introduction to Statistical Modeling of Extreme Values

Author Original S functions by Stuart Coles, R port and R documentation files by Alec Stephenson.

Maintainer Eric Gilleland <ericg@ucar.edu>

Depends R (>= 1.5.0)

Description Functions to support the computations carried out in ‘An Introduction to Statistical Modeling of Extreme Values’ by Stuart Coles. The functions may be divided into the following groups; maxima/minima, order statistics, peaks over thresholds and point processes.

License GPL (>= 2)

URL http://www.ral.ucar.edu/~ericg/softextreme.php

Repository CRAN

Date/Publication 2009-07-18 16:11:51

R topics documented:

`dowjones` *Daily Closing Prices of The Dow Jones Index*

Description

The `dowjones` data frame has 1304 rows and 2 columns. The second column contains daily closing prices of the Dow Jones Index over the period 1996 to 2000. The first column contains a `POSIXct` object giving the dates of each observation.

Usage

```
data(dowjones)
```

Format

This data frame contains the following columns:

Date A `POSIXct` object containing dates.

Index A numeric vector containing daily closing prices of the Dow Jones Index.

Source

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

engine — *Engine Failure Time Data*

Description

The `engine` data frame has 32 rows and 2 columns. The first column contains the corrosion level, the second column gives the engine failure time.

Usage

```
data(engine)
```

Format

This data frame contains the following columns:

Time A numeric vector of corrosion levels.

Corrosion A numeric vector of failure times.

Source

Unknown.

euroex — *UK/Euro Exchange Rates*

Description

A numeric vector of daily exchange rates between the Euro and UK sterling.

Usage

```
data(euroex)
```

Format

A vector containing 975 observations.

Source

Unknown.

`exchange` *UK/US and UK/Canada Exchange Rates*

Description

The `exchange` data frame has 975 rows and 2 columns. The columns contain daily exchange rates; UK sterling against the US dollar (first column) and UK sterling against the Canadian dollar (second column). The rownames contain the corresponding dates in a character string with the format `"2000/05/26"`. This can be converted into a `POSIXct` or `POSIXlt` object using `as.POSIXct` or `as.POSIXlt`.

Usage

```
data(exchange)
```

Format

This data frame contains the following columns:

USD.GBP US against UK exchange rate.

CAD.GBP Canada against UK exchange rate.

Source

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

`fremantle` *Annual Maximum Sea Levels at Fremantle, Western Australia*

Description

The `fremantle` data frame has 86 rows and 3 columns. The second column gives 86 annual maximimum sea levels recorded at Fremantle, Western Australia, within the period 1897 to 1989. The first column gives the corresponding years. The third column gives annual mean values of the Southern Oscillation Index (SOI), which is a proxy for meteorological volitility.

Usage

```
data(fremantle)
```

Format

This data frame contains the following columns:

Year A numeric vector of years.

SeaLevel A numeric vector of annual sea level maxima.

SOI A numeric vector of annual mean values of the Southern Oscillation Index.

Source

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

`gev.diag` *Diagnostic Plots for GEV Models*

Description

Produces diagnostic plots for GEV models using the output of the function `gev.fit`.

Usage

```
gev.diag(z)
```

Arguments

`z`	An object returned by `gev.fit`.

Value

For stationary models four plots are produced; a probability plot, a quantile plot, a return level plot and a histogram of data with fitted density.

For non-stationary models two plots are produced; a residual probability plot and a residual quantile plot.

See Also

`gev.fit`, `gev.prof`

Examples

```
data(portpirie)
ppfit <- gev.fit(portpirie[,2])
gev.diag(ppfit)
```

gev.fit *Maximum-likelihood Fitting of the GEV Distribution*

Description

Maximum-likelihood fitting for the generalized extreme value distribution, including generalized linear modelling of each parameter.

Usage

```
gev.fit(xdat, ydat = NULL, mul = NULL, sigl = NULL, shl = NULL,
    mulink = identity, siglink = identity, shlink = identity,
    muinit = NULL, siginit = NULL, shinit = NULL,
    show = TRUE, method = "Nelder-Mead", maxit = 10000, ...)
```

Arguments

`xdat`	A numeric vector of data to be fitted.
`ydat`	A matrix of covariates for generalized linear modelling of the parameters (or `NULL` (the default) for stationary fitting). The number of rows should be the same as the length of `xdat`.
`mul, sigl, shl`	Numeric vectors of integers, giving the columns of `ydat` that contain covariates for generalized linear modelling of the location, scale and shape parameters repectively (or `NULL` (the default) if the corresponding parameter is stationary).
`mulink, siglink, shlink`	Inverse link functions for generalized linear modelling of the location, scale and shape parameters repectively.
`muinit, siginit, shinit`	numeric of length equal to total number of parameters used to model the location, scale or shape parameter(s), resp. See Details section for default (NULL) initial values.
`show`	Logical; if `TRUE` (the default), print details of the fit.
`method`	The optimization method (see `optim` for details).
`maxit`	The maximum number of iterations.
`...`	Other control parameters for the optimization. These are passed to components of the `control` argument of `optim`.

Details

For non-stationary fitting it is recommended that the covariates within the generalized linear models are (at least approximately) centered and scaled (i.e. the columns of `ydat` should be approximately centered and scaled).

Let m=mean(xdat) and s=sqrt(6*var(xdat))/pi. Then, initial values assigend when 'muinit' is NULL are m - 0.57722 * s (stationary case). When 'siginit' is NULL, the initial value is taken to be s, and when 'shinit' is NULL, the initial value is taken to be 0.1. When covariates are introduced (non-stationary case), these same initial values are used by default for the constant term, and zeros for all other terms. For example, if a GEV(mu(t)=mu0+mu1*t, sigma, xi) is being fitted, then the initial value for mu0 is m - 0.57722 * s, and 0 for mu1.

Value

A list containing the following components. A subset of these components are printed after the fit. If `show` is `TRUE`, then assuming that successful convergence is indicated, the components `nllh`, `mle` and `se` are always printed.

`trans`	An logical indicator for a non-stationary fit.
`model`	A list with components `mul`, `sigl` and `shl`.
`link`	A character vector giving inverse link functions.
`conv`	The convergence code, taken from the list returned by `optim`. A zero indicates successful convergence.
`nllh`	The negative logarithm of the likelihood evaluated at the maximum likelihood estimates.
`data`	The data that has been fitted. For non-stationary models, the data is standardized.
`mle`	A vector containing the maximum likelihood estimates.
`cov`	The covariance matrix.
`se`	A vector containing the standard errors.
`vals`	A matrix with three columns containing the maximum likelihood estimates of the location, scale and shape parameters at each data point.

See Also

`gev.diag`, `optim`, `gev.prof`

Examples

```
data(portpirie)
gev.fit(portpirie[,2])
```

`gev.prof` *Profile Log-likelihoods for Stationary GEV Models*

Description

Produce profile log-likelihoods for shape parameters and m year/block return levels for stationary GEV models using the output of the function `gev.fit`.

Usage

```
gev.prof(z, m, xlow, xup, conf = 0.95, nint = 100)
gev.profxi(z, xlow, xup, conf = 0.95, nint = 100)
```

Arguments

z	An object returned by gev.fit. The object should represent a stationary model.
m	The return level (i.e. the profile likelihood is for the value that is exceeded with probability 1/m).
xlow, xup	The least and greatest value at which to evaluate the profile likelihood.
conf	The confidence coefficient of the plotted profile confidence interval.
nint	The number of points at which the profile likelihood is evaluated.

Value

A plot of the profile likelihood is produced, with a horizontal line representing a profile confidence interval with confidence coefficient conf.

See Also

gev.fit, gev.diag

Examples

```
data(portpirie)
ppfit <- gev.fit(portpirie[,2])
## Not run: gev.prof(ppfit, m = 10, 4.1, 5)
## Not run: gev.profxi(ppfit, -0.3, 0.3)
```

---

glass *Breaking Strengths of Glass Fibres*

Description

A numeric vector containing breaking strengths of 63 glass fibres of length 1.5 centimetres, recorded under experimental conditions.

Usage

```
data(glass)
```

Format

A vector containing 63 observations.

Source

Smith, R. L. and Naylor, J. C. (1987) A comparison of maximum likelihood and Bayesian estimators for the three-parameter Weibull distribution. *Applied Statistics* **36**, 358–396.

References

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

`gpd.diag` *Diagnostic Plots for GPD Models*

Description

Produces diagnostic plots for GPD models using the output of the function `gpd.fit`.

Usage

```
gpd.diag(z)
```

Arguments

`z`	An object returned by `gpd.fit`.

Value

For stationary models four plots are produced; a probability plot, a quantile plot, a return level plot and a histogram of data with fitted density.

For non-stationary models two plots are produced; a residual probability plot and a residual quantile plot.

See Also

`gpd.fit`, `gpd.prof`, `pp.fit`

Examples

```
data(rain)
rnfit <- gpd.fit(rain, 10)
gpd.diag(rnfit)
```

gpd.fit *Maximum-likelihood Fitting for the GPD Model*

Description

Maximum-likelihood fitting for the GPD model, including generalized linear modelling of each parameter.

Usage

```
gpd.fit(xdat, threshold, npy = 365, ydat = NULL, sigl = NULL,
    shl = NULL, siglink = identity, shlink = identity, siginit = NULL,
    shinit = NULL, show = TRUE,
    method = "Nelder-Mead", maxit = 10000, ...)
```

Arguments

`xdat` A numeric vector of data to be fitted.

`threshold` The threshold; a single number or a numeric vector of the same length as `xdat`.

`npy` The number of observations per year/block.

`ydat` A matrix of covariates for generalized linear modelling of the parameters (or `NULL` (the default) for stationary fitting). The number of rows should be the same as the length of `xdat`.

`sigl, shl` Numeric vectors of integers, giving the columns of `ydat` that contain covariates for generalized linear modelling of the scale and shape parameters repectively (or `NULL` (the default) if the corresponding parameter is stationary).

`siglink, shlink` Inverse link functions for generalized linear modelling of the scale and shape parameters repectively.

`siginit, shinit` numeric giving initial value(s) for parameter estimates. If NULL, default is sqrt(6 * var(xdat))/pi and 0.1 for the scale and shape parameters, resp. If using parameter covariates, then these values are used for the constant term, and zeros for all other terms.

`show` Logical; if `TRUE` (the default), print details of the fit.

`method` The optimization method (see `optim` for details).

`maxit` The maximum number of iterations.

`...` Other control parameters for the optimization. These are passed to components of the `control` argument of `optim`.

Details

For non-stationary fitting it is recommended that the covariates within the generalized linear models are (at least approximately) centered and scaled (i.e. the columns of `ydat` should be approximately centered and scaled).

Value

A list containing the following components. A subset of these components are printed after the fit. If `show` is `TRUE`, then assuming that successful convergence is indicated, the components `nexc`, `nllh`, `mle`, `rate` and `se` are always printed.

- `trans` — An logical indicator for a non-stationary fit.
- `model` — A list with components `sigl` and `shl`.
- `link` — A character vector giving inverse link functions.
- `threshold` — The threshold, or vector of thresholds.
- `nexc` — The number of data points above the threshold.
- `data` — The data that lie above the threshold. For non-stationary models, the data is standardized.
- `conv` — The convergence code, taken from the list returned by `optim`. A zero indicates successful convergence.
- `nllh` — The negative logarithm of the likelihood evaluated at the maximum likelihood estimates.
- `vals` — A matrix with three columns containing the maximum likelihood estimates of the scale and shape parameters, and the threshold, at each data point.
- `mle` — A vector containing the maximum likelihood estimates.
- `rate` — The proportion of data points that lie above the threshold.
- `cov` — The covariance matrix.
- `se` — A vector containing the standard errors.
- `n` — The number of data points (i.e. the length of `xdat`).
- `npy` — The number of observations per year/block.
- `xdata` — The data that has been fitted.

See Also

`gpd.diag`, `optim`, `gpd.prof`, `gpd.fitrange`, `mrl.plot`, `pp.fit`

Examples

```
data(rain)
gpd.fit(rain, 10)
```

`gpd.fitrange` *Fitting the GPD Model Over a Range of Thresholds*

Description

Maximum-likelihood fitting for a stationary GPD model, over a range of thresholds. Graphs of parameter estimates which aid the selection of a threshold are produced.

Usage

```
gpd.fitrange(data, umin, umax, nint = 10, show = FALSE)
```

Arguments

`data`	A numeric vector of data to be fitted.
`umin, umax`	The minimum and maximum thresholds at which the model is fitted.
`nint`	The number of fitted models.
`show`	Logical; if `TRUE`, print details of each fit.

Value

Two graphs showing maximum likelihood estimates and confidence intervals of the shape and modified scale parameters over a range of thresholds are produced.

See Also

`gpd.fit`, `mrl.plot`, `pp.fit`, `pp.fitrange`

Examples

```
## Not run: data(rain)
## Not run: gpd.fitrange(rain, 10, 40)
```

`gpd.prof` *Profile Log-likelihoods for Stationary GPD Models*

Description

Produce profile log-likelihoods for shape parameters and m year/block return levels for stationary GPD models using the output of the function `gpd.fit`.

Usage

```
gpd.prof(z, m, xlow, xup, npy = 365, conf = 0.95, nint = 100)
gpd.profxi(z, xlow, xup, conf = 0.95, nint = 100)
```

Arguments

z	An object returned by gpd.fit. The object should represent a stationary model.
m	The return level (i.e. the profile likelihood is for the value that is exceeded with probability 1/m).
xlow, xup	The least and greatest value at which to evaluate the profile likelihood.
npy	The number of observations per year.
conf	The confidence coefficient of the plotted profile confidence interval.
nint	The number of points at which the profile likelihood is evaluated.

Value

A plot of the profile likelihood is produced, with a horizontal line representing a profile confidence interval with confidence coefficient conf.

See Also

gpd.fit, gpd.diag

Examples

```
data(rain)
rnfit <- gpd.fit(rain, 10)
## Not run: gpd.prof(rnfit, m = 10, 55, 75)
## Not run: gpd.profxi(rnfit, -0.02, 0.15)
```

---

gum.diag *Diagnostic Plots for Gumbel Models*

Description

Produces diagnostic plots for Gumbel models using the output of the function gum.fit.

Usage

```
gum.diag(z)
```

Arguments

z	An object returned by gum.fit.

Value

For stationary models four plots are produced; a probability plot, a quantile plot, a return level plot and a histogram of data with fitted density.

For non-stationary models two plots are produced; a residual probability plot and a residual quantile plot.

See Also

gev.fit, gum.fit

Examples

```
data(portpirie)
ppfit <- gum.fit(portpirie[,2])
gum.diag(ppfit)
```

---

gum.fit *Maximum-likelihood Fitting of the Gumbel Distribution*

Description

Maximum-likelihood fitting for the gumbel distribution, including generalized linear modelling of each parameter.

Usage

```
gum.fit(xdat, ydat = NULL, mul = NULL, sigl = NULL, mulink = identity,
    siglink = identity, muinit = NULL, siginit = NULL, show = TRUE,
    method = "Nelder-Mead", maxit = 10000, ...)
```

Arguments

`xdat`	A numeric vector of data to be fitted.
`ydat`	A matrix of covariates for generalized linear modelling of the parameters (or `NULL` (the default) for stationary fitting). The number of rows should be the same as the length of `xdat`.
`mul, sigl`	Numeric vectors of integers, giving the columns of `ydat` that contain covariates for generalized linear modelling of the location and scale parameters repectively (or `NULL` (the default) if the corresponding parameter is stationary).
`mulink, siglink`	Inverse link functions for generalized linear modelling of the location and scale parameters repectively.
`muinit, siginit`	numeric giving initial parameter estimates. See Details section for information about default values (NULL).
`show`	Logical; if `TRUE` (the default), print details of the fit.
`method`	The optimization method (see `optim` for details).
`maxit`	The maximum number of iterations.
`...`	Other control parameters for the optimization. These are passed to components of the `control` argument of `optim`.

Details

For non-stationary fitting it is recommended that the covariates within the generalized linear models are (at least approximately) centered and scaled (i.e. the columns of `ydat` should be approximately centered and scaled).

Let m=mean(xdat) and s=sqrt(6*var(xdat))/pi. Then, initial values assigend when 'muinit' is NULL are m - 0.57722 * s (stationary case). When 'siginit' is NULL, the initial value is taken to be s, and when 'shinit' is NULL. When covariates are introduced (non-stationary case), these same initial values are used by default for the constant term, and zeros for all other terms. For example, if a Gumbel(mu(t)=mu0+mu1*t, sigma) is being fitted, then the initial value for mu0 is m - 0.57722 * s, and 0 for mu1.

Value

A list containing the following components. A subset of these components are printed after the fit. If `show` is `TRUE`, then assuming that successful convergence is indicated, the components `nllh`, `mle` and `se` are always printed.

`trans`	An logical indicator for a non-stationary fit.
`model`	A list with components `mul` and `sigl`.
`link`	A character vector giving inverse link functions.
`conv`	The convergence code, taken from the list returned by `optim`. A zero indicates successful convergence.
`nllh`	The negative logarithm of the likelihood evaluated at the maximum likelihood estimates.
`data`	The data that has been fitted. For non-stationary models, the data is standardized.
`mle`	A vector containing the maximum likelihood estimates.
`cov`	The covariance matrix.
`se`	A vector containing the standard errors.
`vals`	A matrix with two columns containing the maximum likelihood estimates of the location and scale parameters at each data point.

See Also

`gum.diag`, `optim`, `gev.fit`

Examples

```
data(portpirie)
gum.fit(portpirie[,2])
```

ismev — *ismev – The Extremes Toolkit: Weather and Climate Applications of Extreme-Value Statistics*

Description

ismev includes functions to support the computations carried out in ‘An Introduction to Statistical Modeling of Extreme Values’ by Stuart Coles. The functions may be divided into the following groups; maxima/minima, order statistics, peaks over thresholds and point processes. **ismev** is an R port of Stuart Coles’ S-Plus extreme value statistical routines.

Primary functions include: ’gev.fit’, ’gev.diag’, ’gpd.fit’, ’gpd.diag’, ’pp.fit’ and ’pp.diag’.

Original R port was carried out by alec Stephenson, and the package is currently being maintained by Eric Gilleland.

mrl.plot — *Mean Residual Life Plot*

Description

An empirical mean residual life plot, including confidence intervals, is produced. The mean residual life plot aids the selection of a threshold for the GPD or point process models.

Usage

```
mrl.plot(data, umin = min(data), umax = max(data) - 0.1,
    conf = 0.95, nint = 100)
```

Arguments

`data`	A numeric vector of data to be fitted.
`umin, umax`	The minimum and maximum thresholds at which the mean residual life function is calculated.
`conf`	The confidence coefficient for the confidence intervals depicted in the plot.
`nint`	The number of points at which the mean residual life function is calculated.

See Also

`gpd.fit`, `gpd.fitrange`, `pp.fit`

Examples

```
data(rain)
mrl.plot(rain)
```

portpirie — *Annual Maximum Sea Levels at Port Pirie, South Australia*

Description

The `portpirie` data frame has 65 rows and 2 columns. The second column gives annual maximimum sea levels recorded at Port Pirie, South Australia, from 1923 to 1987. The first column gives the corresponding years.

Usage

```
data(portpirie)
```

Format

This data frame contains the following columns:

Year A numeric vector of years.

SeaLevel A numeric vector of annual sea level maxima.

Source

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

pp.diag — *Diagnostic Plots for Point Process Models*

Description

Produces diagnostic plots for point process models using the output of the function `pp.fit`.

Usage

```
pp.diag(z)
```

Arguments

`z`	An object returned by `pp.fit`.

Value

For stationary models two plots are produced; a probability plot and a quantile plot.

For non-stationary models two plots are produced; a residual probability plot and a residual quantile plot.

See Also

pp.fit, gpd.fit

Examples

```
data(rain)
rnfit <- pp.fit(rain, 10)
pp.diag(rnfit)
```

---

pp.fit *Maximum-likelihood Fitting for the Point Process Model*

Description

Maximum-likelihood fitting for the point process model, including generalized linear modelling of each parameter.

Usage

```
pp.fit(xdat, threshold, npy = 365, ydat = NULL, mul = NULL, sigl =
    NULL, shl = NULL, mulink = identity, siglink = identity, shlink =
    identity, muinit = NULL, siginit = NULL, shinit = NULL, show = TRUE,
    method = "Nelder-Mead", maxit = 10000, ...)
```

Arguments

`xdat`	A numeric vector of data to be fitted.
`threshold`	The threshold; a single number or a numeric vector of the same length as `xdat`.
`npy`	The number of observations per year/block.
`ydat`	A matrix of covariates for generalized linear modelling of the parameters (or `NULL` (the default) for stationary fitting). The number of rows should be the same as the length of `xdat`.
`mul, sigl, shl`	Numeric vectors of integers, giving the columns of `ydat` that contain covariates for generalized linear modelling of the location, scale and shape parameters repectively (or `NULL` (the default) if the corresponding parameter is stationary).
`mulink, siglink, shlink`	Inverse link functions for generalized linear modelling of the location, scale and shape parameters repectively.
`muinit, siginit, shinit`	numeric giving initial parameter estimates. See Details section for information on default (NULL) initial values.
`show`	Logical; if `TRUE` (the default), print details of the fit.
`method`	The optimization method (see `optim` for details).
`maxit`	The maximum number of iterations.
`...`	Other control parameters for the optimization. These are passed to components of the `control` argument of `optim`.

Details

For non-stationary fitting it is recommended that the covariates within the generalized linear models are (at least approximately) centered and scaled (i.e. the columns of `ydat` should be approximately centered and scaled). Otherwise, the numerics may become unstable.

As of version 1.32, a more accurate estimate of the exceedance rate, in the face of covariates, is used (at the expense of computational efficiency). In particular, when including covariates, parameter estimates may differ from those in Coles (2001).

Let m=mean(xdat) and s=sqrt(6*var(xdat))/pi. Then, initial values assigend when 'muinit' is NULL are m - 0.57722 * s (stationary case). When 'siginit' is NULL, the initial value is taken to be s, and when 'shinit' is NULL, the initial value is taken to be 0.1. When covariates are introduced (non-stationary case), these same initial values are used by default for the constant term, and zeros for all other terms. For example, if a GEV(mu(t)=mu0+mu1*t, sigma, xi) is being fitted, then the initial value for mu0 is m - 0.57722 * s, and 0 for mu1.

Value

A list containing the following components. A subset of these components are printed after the fit. If `show` is `TRUE`, then assuming that successful convergence is indicated, the components `nexc`, `nllh`, `mle` and `se` are always printed.

`trans`	An logical indicator for a non-stationary fit.
`model`	A list with components `mul`, `sigl` and `shl`.
`link`	A character vector giving inverse link functions.
`threshold`	The threshold, or vector of thresholds.
`npy`	The number of observations per year/block.
`nexc`	The number of data points above the threshold.
`data`	The data that lie above the threshold. For non-stationary models, the data is standardized.
`conv`	The convergence code, taken from the list returned by `optim`. A zero indicates successful convergence.
`nllh`	The negative logarithm of the likelihood evaluated at the maximum likelihood estimates.
`vals`	A matrix with four columns containing the maximum likelihood estimates of the location, scale and shape parameters, and the threshold, at each data point.
`gpd`	A matrix with three rows containing the maximum likelihood estimates of corresponding GPD location, scale and shape parameters at each data point.
`mle`	A vector containing the maximum likelihood estimates.
`cov`	The covariance matrix.
`se`	A vector containing the standard errors.

Warning

Different optimization methods may result in wildly different parameter estimates.

Note

This code is adapted by Eric Gilleland from code originally written for S-Plus by Stuart Coles, and ported to R by Alec Stephenson. See details section above.

References

Beirlant J, Goegebeur Y, Segers J and Teugels J. (2004). Statistics of Extremes, Wiley, Chichester, England.

Coles, Stuart (2001). An Introduction to Statistical Modeling of Extreme Values. Springer-Verlag, London.

See Also

`pp.diag`, `optim`, `pp.fitrange`, `mrl.plot`, `gpd.fit`

Examples

```
data(rain)
pp.fit(rain, 10)
```

`pp.fitrange` *Fitting the Point Process Model Over a Range of Thresholds*

Description

Maximum-likelihood fitting for a stationary point process model, over a range of thresholds. Graphs of parameter estimates which aid the selection of a threshold are produced.

Usage

```
pp.fitrange(data, umin, umax, npy = 365, nint = 10, show = FALSE)
```

Arguments

`data`	A numeric vector of data to be fitted.
`umin, umax`	The minimum and maximum thresholds at which the model is fitted.
`npy`	The number of observations per year/block.
`nint`	The number of fitted models.
`show`	Logical; if `TRUE`, print details of each fit.

Value

Three graphs showing maximum likelihood estimates and confidence intervals of the location, scale and shape parameters over a range of thresholds are produced.

See Also

pp.fit, mrl.plot, gpd.fit, gpd.fitrange

Examples

```
## Not run: data(rain)
## Not run: pp.fitrange(rain, 10, 40)
```

---

rain — *Daily Rainfall Accumulations in South-West England*

Description

A numeric vector containing daily rainfall accumulations at a location in south-west England over the period 1914 to 1962.

Usage

```
data(rain)
```

Format

A vector containing 17531 observations.

Source

Coles, S. G. and Tawn, J. A. (1996) Modelling extremes of the areal rainfall process. *Journal of the Royal Statistical Society, B* **53**, 329–347.

References

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

---

rlarg.diag — *Diagnostic Plots for Order Statistics Models*

Description

Produces diagnostic plots for order statistics models using the output of the function `rlarg.fit`.

Usage

```
rlarg.diag(z, n = z$r)
```

Arguments

`z`	An object returned by `rlarg.fit`.
`n`	Probability and quantile plots are produced for the largest `n` order statistics.

Value

For stationary models four plots are initially produced; a probability plot, a quantile plot, a return level plot and a histogram of data with fitted density. Then probability and quantile plots are produced for the largest `n` order statistics.

For non-stationary models residual probability plots and residual quantile plots are produced for the largest `n` order statistics.

See Also

`rlarg.fit`

Examples

```
## Not run: data(venice)
## Not run: venfit <- rlarg.fit(venice[,-1])
## Not run: rlarg.diag(venfit)
```

`rlarg.fit` *Maximum-likelihood Fitting of Order Statistics Model*

Description

Maximum-likelihood fitting for the order statistic model, including generalized linear modelling of each parameter.

Usage

```
rlarg.fit(xdat, r = dim(xdat)[2], ydat = NULL, mul = NULL, sigl = NULL,
  shl = NULL, mulink = identity, siglink = identity, shlink = identity,
  muinit = NULL, siginit = NULL, shinit = NULL, show = TRUE,
  method = "Nelder-Mead", maxit = 10000, ...)
```

Arguments

`xdat`	A numeric matrix of data to be fitted. Each row should be a vector of decreasing order, containing the largest order statistics for each year (or time period). The first column therefore contains annual (or period) maxima. Only the first `r` columns are used for the fitted model. By default, all columns are used. If one year (or time period) contains fewer order statistics than another, missing values can be appended to the end of the corresponding row.
`r`	The largest `r` order statistics are used for the fitted model.

ydat
: A matrix of covariates for generalized linear modelling of the parameters (or NULL (the default) for stationary fitting). The number of rows should be the same as the number of rows of xdat.

mul, sigl, shl
: Numeric vectors of integers, giving the columns of ydat that contain covariates for generalized linear modelling of the location, scale and shape parameters repectively (or NULL (the default) if the corresponding parameter is stationary).

mulink, siglink, shlink
: Inverse link functions for generalized linear modelling of the location, scale and shape parameters repectively.

muinit, siginit, shinit
: numeric of length equal to total number of parameters used to model the location, scale or shape parameter(s), resp. See Details section for default (NULL) initial values.

show
: Logical; if TRUE (the default), print details of the fit.

method
: The optimization method (see optim for details).

maxit
: The maximum number of iterations.

...
: Other control parameters for the optimization. These are passed to components of the control argument of optim.

Details

For non-stationary fitting it is recommended that the covariates within the generalized linear models are (at least approximately) centered and scaled (i.e. the columns of ydat should be approximately centered and scaled).

Let m=mean(xdat) and s=sqrt(6*var(xdat))/pi. Then, initial values assigend when 'muinit' is NULL are m - 0.57722 * s (stationary case). When 'siginit' is NULL, the initial value is taken to be s, and when 'shinit' is NULL, the initial value is taken to be 0.1. When covariates are introduced (non-stationary case), these same initial values are used by default for the constant term, and zeros for all other terms. For example, if a GEV(mu(t)=mu0+mu1*t, sigma, xi) is being fitted, then the initial value for mu0 is m - 0.57722 * s, and 0 for mu1.

Value

A list containing the following components. A subset of these components are printed after the fit. If show is TRUE, then assuming that successful convergence is indicated, the components nllh, mle and se are always printed.

trans
: An logical indicator for a non-stationary fit.

model
: A list with components mul, sigl and shl.

link
: A character vector giving inverse link functions.

conv
: The convergence code, taken from the list returned by optim. A zero indicates successful convergence.

nllh
: The negative logarithm of the likelihood evaluated at the maximum likelihood estimates.

data
: The data that has been fitted. For non-stationary models, the data is standardized.

`mle`	A vector containing the maximum likelihood estimates.
`cov`	The covariance matrix.
`se`	A vector containing the standard errors.
`vals`	A matrix with three columns containing the maximum likelihood estimates of the location, scale and shape parameters at each data point.
`r`	The number of order statistics used.

See Also

`rlarg.diag`, `optim`

Examples

```
## Not run: data(venice)
## Not run: rlarg.fit(venice[,-1])
```

`venice` *Venice Sea Levels*

Description

The `venice` data frame has 51 rows and 11 columns. The final ten columns contain the 10 largest sea levels observed within the year given by the first column. The ten largest sea levels are given for every year in the period 1931 to 1981, excluding 1935 in which only the six largest measurements are available.

Usage

```
data(venice)
```

Format

This data frame contains the following columns:

Year A numeric vector of years.

r1 Annual sea level maxima.

r2 The second largest sea level.

r3 The third largest sea level.

r4 The forth largest sea level.

r5 The fifth largest sea level.

r6 The sixth largest sea level.

r7 The seventh largest sea level.

r8 The eigth largest sea level.

r9 The ninth largest sea level.

r10 The tenth largest sea level.

Source

Smith, R. L. (1986) Extreme value theory based on the *r* largest annual events. *Journal of Hydrology* **86**, 27–43.

References

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

`wavesurge` *Wave and Surge Heights in South-West England*

Description

The `wavesurge` data frame has 2894 rows and 2 columns. The columns contain wave and surge heights (in metres) at a single location off south-west England.

Usage

```
data(wavesurge)
```

Format

This data frame contains the following columns:

wave A numeric vector of wave heights.

surge A numeric vector of surge heights.

Source

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

`wind` *Annual Maximum Wind Speeds at Albany and Hartford*

Description

The `wind` data frame has 40 rows and 3 columns. The second and third columns contain annual maximum wind speeds at Albany, New York and Hartford, Connecticut respectively, over the period 1944 to 1983. The first column gives the corresponding years.

Usage

```
data(wind)
```

Format

This data frame contains the following columns:

Year A numeric vector of years.

Hartford Annual maximum wind speeds at Hartford.

Albany Annual maximum wind speeds at Albany.

Source

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

`wooster` *Minimum Temperatures at Wooster, Ohio*

Description

A numeric vector containing daily minimum temperatures, in degrees Fahrenheit, at Wooster, Ohio, over the period 1983 to 1988.

Usage

```
data(wooster)
```

Format

A vector containing 1826 observations.

Source

Coles, S. G., Tawn, J. A. and Smith, R. L. (1994) A seasonal Markov model for extremely low temperatures. *Environmetrics* **5**, 221–239.

References

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values.* London: Springer.

Smith, R. L., Tawn, J. A. and Coles, S. G. (1997) Markov chain models for threshold exceedences. *Biometrica* **84**, 249–268.

Index