Package ‘gamlssx’

February 28, 2025

Type Package

Title Generalized Additive Extreme Value Models for Location, Scale and Shape

Version 1.0.2

Date 2025-02-28

Description Fits generalized additive models for the location, scale and shape parameters of a generalized extreme value response distribution. The methodology is based on Rigby, R.A. and Stasinopoulos, D.M. (2005), <doi:10.1111/j.1467-9876.2005.00510.x> and implemented using functions from the 'gamlss' package <doi:10.32614/CRAN.package.gamlss>.

Imports gamlss, gamlss.dist, nieve, stats

License GPL (>= 3)

LazyData TRUE

Encoding UTF-8

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0)

URL `https://paulnorthrop.github.io/gamlssx/`, `https://github.com/paulnorthrop/gamlssx`

BugReports `https://github.com/paulnorthrop/gamlssx/issues`

Config/testthat/edition 3

RoxygenNote 7.3.2

NeedsCompilation no

Author Paul J. Northrop [aut, cre, cph], Jennifer Ji [aut]

Maintainer Paul J. Northrop <`p.northrop@ucl.ac.uk`>

Repository CRAN

Date/Publication 2025-02-28 19:10:02 UTC

Contents

`gamlssx-package` *gamlssx: Generalized Additive Extreme Value Models for Location, Scale and Shape*

Description

Fits generalized additive models for the location, scale and shape parameters of a generalized extreme value response distribution. The methodology is based on Rigby and Stasinopoulos (2005) and implemented using functions from the `gamlss` package doi:10.32614/CRAN.package.gamlss.

Details

The main function in `gamlssx` is `fitGEV()`, which calls the function `gamlss::gamlss()`. See the gamlssx package page on Github for more information.

Author(s)

Maintainer: Paul J. Northrop <`p.northrop@ucl.ac.uk`> [copyright holder]

Authors:

- Jennifer Ji

References

Rigby R.A. and Stasinopoulos D.M. (2005). Generalized additive models for location, scale and shape (with discussion), *Appl. Statist.*, **54**, part 3, pages 507-554. doi:10.1111/j.14679876.2005.00510.x

See Also

`fitGEV()`, `gamlss::gamlss()`

`fitGEV` *Fit a Generalized Extreme value (GEV) GAMLSS Model*

Description

Fits a Generalized Additive Model (GAM) for Location, Scale and Shape with a GEV response distribution, using the function `gamlss::gamlss()`.

Usage

```
fitGEV(
  formula,
  data,
  scoring = c("fisher", "quasi"),
  mu.link = "identity",
  sigma.link = "log",
  xi.link = "identity",
  stepLength = 1,
  stepAttempts = 2,
  stepReduce = 2,
  steps = FALSE,
  ...
)
```

Arguments

`formula` a formula object, with the response on the left of an ~ operator, and the terms, separated by + operators, on the right. Nonparametric smoothing terms are indicated by `pb()` for penalised beta splines, `cs` for smoothing splines, `lo` for `loess` smooth terms and `random` or `ra` for random terms, e.g. `y~cs(x,df=5)+x1+x2*x3`. Additional smoothers can be added by creating the appropriate interface. Interactions with nonparametric smooth terms are not fully supported, but will not produce errors; they will simply produce the usual parametric interaction

`data` a data frame containing the variables occurring in the formula, e.g. `data=aids`. If this is missing, the variables should be on the search list.

`scoring` A character scalar. If `scoring = "fisher"` then the weights used in the fitting algorithm are based on the expected Fisher information, that is, a Fisher's scoring algorithm is used. If `scoring = "quasi"` then these weights are based on the cross products of the first derivatives of the log-likelihood, leading to a quasi Newton scoring algorithm.

`mu.link`, `sigma.link`, `xi.link`
Character scalars to set the respective link functions for the location (`mu`), scale (`sigma`) and shape (`xi`) parameters. The latter is passed to `gamlss::gamlss()` as `nu.link`.

`stepLength`	A numeric vector of positive values. The initial values of the step lengths `mu.step`, `sigma.step` and `nu.step` passed to `gamlss::gamlss.control()` in the first attempt to fit the model by calling `gamlss::gamlss()`. If `stepLength` has a length that is less than 3 then `stepLength` is recycled to have length 3.
`stepAttempts`	A non-negative integer. If the first call to `gamlss::gamlss()` throws an error then we make `stepAttempts` further attempts to fit the model, each time dividing by 2 the values of `mu.step`, `sigma.step` and `nu.step` supplied to `gamlss::gamlss.control()`. If `stepAttempts < 1` then no further attempts are made.
`stepReduce`	A number greater than 1. The factor by which the step lengths in `stepLength` are reduced for each extra attempt to fit the model. The default, `stepReduce = 2` means that the step lengths are halved for each extra attempt.
`steps`	A logical scalar. Pass `steps = TRUE` to write to the console the current value of `stepLength` for each call to `gamlss::gamlss()`.
`...`	Further arguments passed to `gamlss::gamlss()`, in particular `method`, which sets the fitting algorithm, with options `RS()`, `CG()` or `mixed()`. The default, `method = RS()` seems to work well, as does `method = mixed()`. In contrast, `method = CG()` often requires the step length to be reduced before convergence is achieved. `fitGEV()` attempts to do this automatically. See `stepAttempts`. Pass `trace = FALSE` (to `gamlss::gamlss.control()`) to avoid writing to the console the global deviance after each outer iteration of the gamlss fitting algorithm.

Details

See `gamlss::gamlss()` for information about the model and the fitting algorithm.

Value

Returns a `gamlss` object. See the **Value** section of `gamlss::gamlss()`. The class of the returned object is `c("gamlssx", "gamlss", "gam", "glm", "lm")`.

See Also

`GEV`, `gamlss.dist::gamlss.family()`, `gamlss::gamlss()`

Examples

```
# Load gamlss, for the functions term.plot() and wp()
library(gamlss)

##### Simulated data

set.seed(17012023)
n <- 100
x <- stats::runif(n)
mu <- 1 + 2 * x
sigma <- 1
xi <- 0.25
```

```
y <- nieve::rGEV(n = 1, loc = mu, scale = sigma, shape = xi)
data <- data.frame(y = as.numeric(y), x = x)
plot(x, y)

# Fit model using the default RS method with Fisher's scoring
mod <- fitGEV(y ~ pb(x), data = data)
# Summary of model fit
summary(mod)
# Residual diagnostic plots
plot(mod, xlab = "x", ylab = "y")
# Data plus fitted curve
plot(data$x, data$y, xlab = "x", ylab = "y")
lines(data$x, fitted(mod))

# Fit model using the mixed method and quasi-Newton scoring
# Use trace = FALSE to prevent writing the global deviance to the console
mod <- fitGEV(y ~ pb(x), data = data, method = mixed(), scoring = "quasi",
              trace = FALSE)

# Fit model using the CG method
# The default step length of 1 needs to be reduced to enable convergence
# Use steps = TRUE to write the step lengths to the console
mod <- fitGEV(y ~ pb(x), data = data, method = CG(), steps = TRUE)

##### Fremantle annual maximum sea levels
##### See also the gamlssx package README file

# Transform Year so that it is centred on 0
fremantle <- transform(fremantle, cYear = Year - median(Year))

# Plot sea level against year and against SOI
plot(fremantle$Year, fremantle$SeaLevel, xlab = "year", ylab = "sea level (m)")
plot(fremantle$SOI, fremantle$SeaLevel, xlab = "SOI", ylab = "sea level (m)")

# Fit a model with P-spline effects of cYear and SOI on location and scale
# The default links are identity for location and log for scale
mod <- fitGEV(SeaLevel ~ pb(cYear) + pb(SOI),
             sigma.formula = ~ pb(cYear) + pb(SOI),
             data = fremantle)

# Summary of model fit
summary(mod)
# Model diagnostic plots
plot(mod)
# Worm plot
wp(mod)
# Plot of the fitted component smooth functions
# Note: gamlss::term.plot() does not include uncertainty about the intercept
# Location mu
term.plot(mod, rug = TRUE, pages = 1)
# Scale sigma
term.plot(mod, what = "sigma", rug = TRUE, pages = 1)
```

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

Description

This is a copy of the `fremantle` dataset from the `ismev` package. The `fremantle` data frame has 86 rows and 3 columns. The second column gives 86 annual maximum 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 volatility.

Usage

```
fremantle
```

Format

This data frame contains the following:

- 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.

Examples

```
summary(fremantle)
```

`GEV` *GEV family distribution for fitting a GAMLSS*

Description

The functions `GEVfisher()` and `GEVquasi()` each define the generalized extreme value (GEV) family distribution, a three parameter distribution, for a `gamlss.dist::gamlss.family()` object to be used in GAMLSS fitting using the function `gamlss::gamlss()`. The only difference between `GEVfisher()` and `GEVquasi()` is the form of scoring method used to define the weights used in the fitting algorithm. Fisher's scoring, based on the expected Fisher information is used in `GEVfisher()`, whereas a quasi-Newton scoring, based on the cross products of the first derivatives of the log-likelihood, is used in `GEVquasi()`. The functions `dGEV`, `pGEV`, `qGEV` and `rGEV` define the density, distribution function, quantile function and random generation for the specific parameterization of the generalized extreme value distribution given in **Details** below.

Usage

```
GEVfisher(mu.link = "identity", sigma.link = "log", nu.link = "identity")

GEVquasi(mu.link = "identity", sigma.link = "log", nu.link = "identity")

dGEV(x, mu = 0, sigma = 1, nu = 0, log = FALSE)

pGEV(q, mu = 0, sigma = 1, nu = 0, lower.tail = TRUE, log.p = FALSE)

qGEV(p, mu = 0, sigma = 1, nu = 0, lower.tail = TRUE, log.p = FALSE)

rGEV(n, mu = 0, sigma = 1, nu = 0)
```

Arguments

`mu.link`	Defines the `mu.link`, with `"identity"` link as the default for the `mu` parameter.
`sigma.link`	Defines the `sigma.link`, with `"log"` link as the default for the `sigma` parameter.
`nu.link`	Defines the `nu.link`, with `"identity"` link as the default for the `nu` parameter.
`x, q`	Vector of quantiles.
`mu, sigma, nu`	Vectors of location, scale and shape parameter values.
`log, log.p`	Logical. If `TRUE`, probabilities `eqn{p}` are given as $\log(p)$.
`lower.tail`	Logical. If `TRUE` (the default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
`p`	Vector of probabilities.
`n`	Number of observations. If `length(n)` > 1, the length is taken to be the number required.

Details

The distribution function of a GEV distribution with parameters `location` = μ, `scale` = $\sigma (> 0)$ and `shape` = ξ $(= \nu)$ is

$$F(x \mid \mu, \sigma, \xi) = P(X \leq x) = \exp\left\{ -\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]_{+}^{-1/\xi} \right\},$$

where $x_{+} = \max(x, 0)$. If $\xi = 0$ the distribution function is defined as the limit as ξ tends to zero. The support of the distribution depends on ξ: it is $x \leq \mu - \sigma/\xi$ for $\xi < 0$; $x \geq \mu - \sigma/\xi$ for $\xi > 0$; and x is unbounded for $\xi = 0$. See `https://en.wikipedia.org/wiki/Generalized_extreme_value_distribution` and/or Chapter 3 of Coles (2001) for further information.

For each observation in the data, the restriction that $\xi > -1/2$ is imposed, which is necessary for the usual asymptotic likelihood theory to be applicable.

Value

`GEVfisher()` and `GEVquasi()` each return a `gamlss.dist::gamlss.family()` object which can be used to fit a regression model with a GEV response distribution using the `gamlss::gamlss()` function. `dGEV()` gives the density, `pGEV()` gives the distribution function, `qGEV()` gives the quantile function, and `rGEV()` generates random deviates.

Examples

See the examples in `fitGEV()`.

References

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London. Chapter 3: doi:10.1007/9781447136750_3

See Also

`fitGEV`, `gamlss.dist::gamlss.family()`, `gamlss::gamlss()`

`gevExpInfo` *GEV Distribution Expected Information*

Description

Calculates the expected information matrix for the GEV distribution.

Usage

```
gev11e(scale, shape)

gev22e(scale, shape, eps = 0.003)

gev33e(shape, eps = 0.003)

gev12e(scale, shape, eps = 0.003)

gev13e(scale, shape, eps = 0.003)

gev23e(scale, shape, eps = 0.003)

gevExpInfo(scale, shape, eps = 0.003)
```

Arguments

`scale, shape`	Numeric vectors. Respective values of the GEV parameters scale parameter σ and shape parameter ξ. For `gevExpInfo`, `scale` and `shape` must have length 1.
`eps`	A numeric scalar. For values of ξ in shape that lie in `(-eps, eps)` an approximation is used instead of a direct calculation. See **Details**. If eps is a vector then only the first element is used.

Details

`gevExpInfo` calculates, for a single pair of values $(\sigma, \xi) =$ `(scale, shape)`, the expected information matrix for a single observation from a GEV distribution with distribution function

$$F(x) = P(X \leq x) = \exp\left\{ -\left[1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right]_+^{-1/\xi} \right\},$$

where $x_+ = \max(x, 0)$. The GEV expected information is defined only for $\xi > -0.5$ and does not depend on the value of μ.

The other functions are vectorized and calculate the individual contributions to the expected information matrix. For example, `gev11e` calculates the expectation $i_{\mu\mu}$ of the negated second derivative of the GEV log-density with respect to μ, that is, each `1` indicates one derivative with respect to μ. Similarly, `2` denotes one derivative with respect to σ and `3` one derivative with respect to ξ, so that, for example, `gev23e` calculates the expectation $i_{\sigma\xi}$ of the negated GEV log-density after one taking one derivative with respect to σ and one derivative with respect to ξ. Note that $i_{\xi\xi}$, calculated using `gev33e`, depends only on ξ.

The expectation in `gev11e` can be calculated in a direct way for all $\xi > -0.5$. For the other components, direct calculation of the expectation is unstable when ξ is close to 0. Instead, we use a quadratic approximation over `(-eps, eps)`, from a Lagrangian interpolation of the values from the direct calculation for $\xi =$ `-eps`, 0 and `eps`.

Value

`gevExpInfo` returns a 3 by 3 numeric matrix with row and column named `loc, scale, shape`. The other functions return a numeric vector of length equal to the maximum of the lengths of the arguments, excluding `eps`.

Examples

```
# Expected information matrices for ...
# ... scale = 2 and shape = -0.4
gevExpInfo(2, -0.4)
# ... scale = 3 and shape = 0.001
gevExpInfo(3, 0.001)
# ... scale = 3 and shape = 0
gevExpInfo(3, 0)
# ... scale = 1 and shape = 0.1
gevExpInfo(1, 0.1)

# The individual components of the latter matrix
gev11e(1, 0.1)
gev12e(1, 0.1)
gev13e(1, 0.1)
gev22e(1, 0.1)
gev23e(1, 0.1)
gev33e(0.1)
```

Index