Generalised Extreme Value (GEV) distribution

Parametrisation

The GEV distribution is defined through the cummulative distribution function

$$F(y;\eta,\tau,\xi) = \exp\left(-\left[1+\xi\sqrt{\tau s}(y-\eta)\right]^{-1/\xi}\right)$$

for

$$1+\xi\sqrt{\tau s}(y-\eta) > 0$$

and for a continuously response y where

η: is the linear predictor

τ: is the "precision"

s: is a fixed scaling, $s > 0$.

Link-function

The linear predictor is given in the parameterisation of the GEV distribution.

Hyperparameters

The GEV-models has two hyperparameters. The "precision" is represented as

$$\theta_1 = \log\tau$$

and the prior is defined on θ_1. The shape parameter ξ is represented as

$$\theta_2 = \xi$$

and the prior is defined on θ_2. [1]

Specification

- family = `gev`
- Required arguments: y and s (keyword `scale`)
- The scaling ξ_s is given by the argument `gev.scale.xi` and is default set to 0.01.

The weights has default value 1.

[1] Internally, the parameter θ_2 is scaled with a fixed scaling ξ_s (default 0.01), to improve the numerics as the natural "scale" of ξ is small. For this reason the $\theta_2(=\xi)$ reported in `result$mode$theta` will appear as θ_2/ξ_s. For the same reason, if you define the mode using `control.mode = list(theta = ..., ...)` then the element representing θ_2 should be given as θ_2/ξ_s.

Hyperparameter spesification and default values

hyper

theta1

name log precision

short.name prec

initial 4

fixed FALSE

prior loggamma

param 1 5e-05

to.theta `function(x) log(x)`

from.theta `function(x) exp(x)`

theta2

name gev parameter

short.name gev

initial 0

fixed FALSE

prior gaussian

param 0 25

to.theta `function(x) x`

from.theta `function(x) x`

survival FALSE

discrete FALSE

link default identity

status experimental

pdf gev

Example

In the following example, we estimate the parameters of the GEV distribution on some simulated data.

```
rgev = function(n=1, xi = 0, mu = 0.0, sd = 1.0) {
    u = runif(n)
    if (xi == 0) {
        x = -log(-log(u))
    } else {
        x = ((-log(u))^(-xi) - 1.0)/xi
    }
    return (x*sd + mu)
}

n = 300
z = rnorm(n)
sd.y = 0.5
```

```
xi = 0.2
y = 1+z + rgev(n, xi=xi, sd = sd.y)

r = inla(y ~ 1 + z, data = data.frame(y, z), family = "gev",
        control.family = list(gev.scale.xi = 0.01))
summary(r)
```

Notes

None.