

# 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

$\eta$ : is the linear predictor

$\tau$ : 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  $\theta_1$ . The shape parameter  $\xi$  is represented as

$$\xi = \xi_s \theta_2$$

where  $\xi_s > 0$  is a *chosen fixed scaling*, and the prior is defined on  $\theta_2$ <sup>1</sup>

## Specification

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

The weights has default value 1.

## Hyperparameter spesification and default values

**hyper**

**theta1**

**name** log precision

**short.name** prec

**initial** 4

**fixed** FALSE

---

<sup>1</sup>The  $\xi_s$  parameter is there for numerical reasons only, as the natural “scale” of  $\xi$  is small, and the scaling makes the natural scale of  $\theta_2$  similar to other  $\theta$ ’s. The output from INLA reports the parameter  $\xi$ .

```

    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 6.25
    to.theta function(x) x
    from.theta function(x) x

survival FALSE

discrete FALSE

link default identity

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 = 100
z = rnorm(n)
sd.y = 0.5
xi = 0
y = 1+z + rgev(n, xi=xi, sd = sd.y)

formula = y ~ 1 + f(inla.group(z), model="rw1")
data = data.frame(y,z)

r = inla(formula, data = data, family = "gev",
  control.family = list(gev.scale.xi = 0.01,
    ## just to show how to set an initial value
    hyper = list(prec=list(initial=2))))

```

## Notes

None.