

# LogLogistic likelihood

## Parametrisation

The LogLogistic distribution has cumulative distribution function

$$F_0(y) = \frac{1}{1 + \lambda y^{-\alpha}}, \quad y > 0$$

if `variant=0`, or

$$F_1(y) = \frac{1}{1 + (\lambda y)^{-\alpha}}, \quad y > 0$$

if `variant=1`, where

$\alpha > 0$  is a shape parameter, and

$\lambda > 0$  is a scale parameter.

## Link-functions

The parameter  $\lambda$  is linked to the linear predictor, by default as

$$\lambda = \exp(\eta)$$

## Hyperparameters

The  $\alpha$  parameter is represented as

$$\theta = \log \alpha$$

and the prior is defined on  $\theta$ .

## Specification

- `family="loglogistic"` (regression) or `"loglogistic.surv"` (survival)
- `variant=0` (default) or 1, choosing between parameterisation  $F_0$  or  $F_1$ .
- Required arguments:  $y$  (regression) or an `inla.surv`-object using `inla.surv()` (for survival data)

## Hyperparameter specification and default values

### Regression:

**doc** The loglogistic likelihood

**hyper**

**theta**

**hyperid** 80001

**name** log alpha

**short.name** alpha

**initial** 1

**fixed** FALSE

**prior** loggamma

**param** 25 25

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

**survival** FALSE

**discrete** FALSE

**link** default log neglog

**pdf** loglogistic

**Survival:**

**doc** The loglogistic likelihood (survival)

**hyper**

**theta1**

```
hyperid 80011
name log alpha
short.name alpha
initial 1
fixed FALSE
prior loggamma
param 25 25
to.theta function(x) log(x)
from.theta function(x) exp(x)
```

**theta2**

```
hyperid 80012
name beta1
short.name beta1
initial -5
fixed FALSE
prior normal
param -4 100
to.theta function(x) x
from.theta function(x) x
```

**theta3**

```
hyperid 80013
name beta2
short.name beta2
initial 0
fixed FALSE
prior normal
param 0 100
to.theta function(x) x
from.theta function(x) x
```

**theta4**

```

hyperid 80014
name beta3
short.name beta3
initial 0
fixed FALSE
prior normal
param 0 100
to.theta function(x) x
from.theta function(x) x
theta5
hyperid 80015
name beta4
short.name beta4
initial 0
fixed FALSE
prior normal
param 0 100
to.theta function(x) x
from.theta function(x) x
theta6
hyperid 80016
name beta5
short.name beta5
initial 0
fixed FALSE
prior normal
param 0 100
to.theta function(x) x
from.theta function(x) x
theta7
hyperid 80017
name beta6
short.name beta6
initial 0
fixed FALSE
prior normal
param 0 100
to.theta function(x) x
from.theta function(x) x
theta8
hyperid 80018
name beta7
short.name beta7
initial 0

```

```

    fixed FALSE
    prior normal
    param 0 100
    to.theta function(x) x
    from.theta function(x) x
theta9
    hyperid 80019
    name beta8
    short.name beta8
    initial 0
    fixed FALSE
    prior normal
    param 0 100
    to.theta function(x) x
    from.theta function(x) x
theta10
    hyperid 80020
    name beta9
    short.name beta9
    initial 0
    fixed FALSE
    prior normal
    param 0 100
    to.theta function(x) x
    from.theta function(x) x
theta11
    hyperid 80021
    name beta10
    short.name beta10
    initial 0
    fixed FALSE
    prior normal
    param 0 100
    to.theta function(x) x
    from.theta function(x) x

survival TRUE

discrete FALSE

link default log neglog

pdf loglogistic

```

## Example

In the following example we estimate the parameters in a simulated case

```
rloglogistic = function(n, lambda, alpha, variant=0)
{
  u = runif(n)
  if (variant == 0) {
    y = (lambda/(1.0/u - 1.0))^(1.0/alpha)
  } else if (variant == 1) {
    y = (1.0/(1.0/u - 1.0))^(1.0/alpha) / lambda
  } else {
    stop("ERROR")
  }
}

n = 1000
alpha = 2.1
x = c(scale(runif(n)))
eta = 1.1+2.2*x
lambda = exp(eta)

for(variant in 0:1) {

  print(paste("variant=", variant))
  y = rloglogistic(n, lambda = lambda,
                  alpha = alpha,
                  variant = variant)

  formula = y ~ 1 + x
  r=inla(formula,
        family = "loglogistic",
        data=data.frame(y, x),
        control.family = list(variant = variant))
  print("REGRESSION")
  print(summary(r))

  event = rep(1,n)
  formula=inla.surv(y,event) ~ 1 + x
  r=inla(formula,
        family = "loglogisticsurv",
        data = list(y=y, event=event, x=x),
        control.family = list(variant = variant))
  print("SURVIVAL")
  print(summary(r))
}
```

## Notes

- Loglogisticsurv model can be used for right censored, left censored, interval censored data. If the observed times  $y$  are large/huge, then this can cause numerical overflow in the likelihood routine. If you encounter this problem, try to scale the observations, `time = time / max(time)`

or similar.