

# LogNormal

## Parametrisation

The LogNormal has density

$$f(y) = \frac{1}{y\sqrt{2\pi}} \sqrt{\tau} \exp\left(-\frac{1}{2}\tau(\log y - \mu)^2\right), \quad y > 0$$

where

$\tau > 0$  is the precision parameter,

$\mu$  is the mean parameter.

## Link-function

The parameter  $\mu$  is linked to the linear predictor as:

$$\eta = \mu$$

## Hyperparameters

The  $\tau$  parameter is represented as

$$\theta = \log \tau$$

and the prior is defined on  $\theta$ .

## Specification

- family = `lognormal` for regression models and family = `lognormalsurv` for survival models.
- Required arguments:  $y$  Given in a format by using `inla.surv()` function for family = `lognormal.surv`

## Hyperparameter spesification and default values

### `lognormal`

#### `hyper`

##### `theta`

`hyperid` 77101

`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)

`survival` FALSE

`discrete` FALSE

`link` default identity

`pdf` lognormal

## lognormalsurv

```
hyper
  theta
    hyperid 78001
    name log precision
    short.name prec
    initial 2
    fixed FALSE
    prior loggamma
    param 1 5e-05
    to.theta function(x) log(x)
    from.theta function(x) exp(x)
survival TRUE
discrete FALSE
link default identity
pdf lognormal
```

## Example

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

```
## these should give the same results
n = 300
x = runif(n)
eta = 1+x
y = exp(rnorm(n, mean = eta, sd = 1))
data = list(y=y, event=rep(1, n), x=x)
formula = inla.surv(y, event) ~ 1 + x
r=inla(formula, family ="lognormalsurv", data=data)
summary(r)

data = data.frame(y, x)
formula = y ~ 1 + x
r=inla(formula, family ="lognormal", data=data)
summary(r)
```

## Notes

- lognormalsurv can be used for right censored, left censored, interval censored data. A general framework to represent time is given by `inla.surv`. If the observed times  $y$  are large/huge, then this can cause numerical overflow, and if you encounter this problem, try to scale the observations, like `time = time / max(time)`.