LogNormal

Parametrisation

The LogNormal has density

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

where

$\tau > 0$ is the precision parameter,

μ is the mean parameter.

Link-function

The parameter μ is linked to the linear predictor as:

$$\eta = \mu$$

Hyperparameters

The τ parameter is represented as

$$\theta = \log\tau$$

and the prior is defined on θ.

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

> **doc** The log-Normal likelihood
>
> **hyper**
>> **theta**
>>> **hyperid** 77101
>>> **name** log precision
>>> **short.name** prec
>>> **initial** 0
>>> **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

>**doc** The log-Normal likelihood (survival)
>
>**hyper**
>
>>**theta**
>>
>>>**hyperid** 78001
>>>
>>>**name** log precision
>>>
>>>**short.name** prec
>>>
>>>**initial** 0
>>>
>>>**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

```
n = 300
x = c(scale(runif(n)))
eta = 1+2.2*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 observatios, like `time = time / max(time)`.