# The Gamma-distribution

## Parametrisation

The Gamma-distribution has the following density

$$\pi(y) = \frac{b^a}{\Gamma(a)} y^{a-1} \exp(-by), \qquad a > 0, \quad b > 0, \quad y > 0,$$

where $\mathrm{E}(y) = \mu = a/b$ and $\mathrm{Var}(y) = 1/\tau = a/b^2$, where $\tau$ is the precision and $\mu$ is the mean. We will use the following parameterisation for the precision

$$\tau = (s\phi)/\mu^2$$

where $\phi$ is the precision parameter (or $1/\phi$ is the dispersion parameter) and $s > 0$ is a fixed scaling, which gives this density

$$\pi(y) = \frac{1}{\Gamma(s\phi)} \left(\frac{(s\phi)}{\mu}\right)^{(s\phi)} y^{(s\phi)-1} \exp\left(-(s\phi)\frac{y}{\mu}\right)$$

## Link-function

The linear predictor $\eta$ is linked to the mean $\mu$ using a default log-link

$$\mu = \exp(\eta)$$

## Hyperparameter

The hyperparameter is the precision parameter $\phi$, which is represented as

$$\phi = \exp(\theta)$$

and the prior is defined on $\theta$.

## Specification

- family = `gamma` for regression models and family = `gamma.surv` for survival models.
- Required arguments: for `gamma.surv`, $y$ (to be given in a format by using `inla.surv()`), and for `gamma`, $y$ and $s$.

The scalings have default value 1.

### Hyperparameter spesification and default values

**doc** The Gamma likelihood

**hyper**

**theta**

**hyperid** 58001
**name** precision parameter
**short.name** prec
**initial** 4.60517018598809
**fixed** FALSE

**prior** loggamma
**param** 1 0.01
**to.theta** `function(x) log(x)`
**from.theta** `function(x) exp(x)`

**survival** FALSE

**discrete** FALSE

**link** default log quantile

**pdf** gamma

**doc** The Gamma likelihood (survival)

**hyper**

**theta1**

**hyperid** 58101
**name** precision parameter
**short.name** prec
**initial** 0
**fixed** FALSE
**prior** loggamma
**param** 1 0.01
**to.theta** `function(x) log(x)`
**from.theta** `function(x) exp(x)`

**theta2**

**hyperid** 58102
**name** beta1
**short.name** beta1
**initial** -7
**fixed** FALSE
**prior** normal
**param** -4 100
**to.theta** `function(x) x`
**from.theta** `function(x) x`

**theta3**

**hyperid** 58103
**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** 58104
**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** 58105
**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** 58106
**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** 58107
**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** 58108
**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** 58109
**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** 58110
**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** 58111
**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

**status** experimental

**link** default log neglog

**pdf** gammasurv

## Example

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

```
n = 1000
x = rnorm(n)
eta = 1 + x
mu = exp(eta)
prec.scale = runif(n, min = 0.5, max = 2)
prec.par = 1.2
a = prec.par * prec.scale
b = mu / (prec.par * prec.scale)
y = rgamma(n, shape = a, scale = b)
r = inla(y ~ 1 + x,  data = data.frame(y, x),
        scale = prec.scale, family = "gamma")
```

## Notes

None.