

## Binomial

### Parametrisation

The Binomial distribution is

$$\text{Prob}(y) = \binom{n}{y} p^y (1-p)^{n-y}$$

for responses  $y = 0, 1, 2, \dots, n$ , where

$n$ : number of trials.

$p$ : probability of success in each trial.

### Link-function

The mean and variance of  $y$  are given as

$$\mu = np \quad \text{and} \quad \sigma^2 = np(1-p)$$

and the probability  $p$  is linked to the linear predictor by

$$p(\eta) = \frac{\exp(\eta)}{1 + \exp(\eta)}$$

### Hyperparameters

None.

### Hyperparameter specification and default values

**doc** The Binomial likelihood

**hyper**

**survival** FALSE

**discrete** TRUE

**link** default logit loga cauchit probit cloglog loglog log sslogit logitoffset quantile pquantile robit sn

**pdf** binomial

### Specification

- family = `binomial`
- Required arguments:  $y$  and  $n$  (keyword `Ntrials`)

### Expert version

There is also an “expert” version were you are supposed to know what you are doing. Here, we allow  $y$  and  $n$  to be non-integers, however, the condition  $0 \leq y \leq n$  apply. The normalizing constant is computed as above using the integer part of  $y$  and  $n$ . This is similar to using `floor(y)` and `floor(n)` in R. The marginal likelihood estimate will in this case make less sense.

- family = `xbinomial`
- Required arguments:  $y$  and  $n$  (keyword `Ntrials`)

**doc** The Binomial likelihood (expert version)

**hyper**

**survival** FALSE

**discrete** TRUE

**link** default logit loga cauchit probit cloglog loglog log sslogit logitoffset quantile pquantile robit sn

**pdf** binomial

**status** experimental

## Example

In the following example we estimate the parameters in a simulated example with binomial responses.

```
n=100
a = 1
b = 1
z = rnorm(n)
eta = a + b*z
Ntrials = sample(c(1,5,10,15), size=n, replace=TRUE)
prob = exp(eta)/(1 + exp(eta))
y = rbinom(n, size=Ntrials, prob = prob)

data = list(y=y,z=z)
formula = y ~ 1+z
result = inla(formula, family = "binomial", data = data, Ntrials=Ntrials)
summary(result)
```

## Notes

If the response is a **factor** it must be converted to  $\{0,1\}$  before calling `inla()`, as this conversion is not done automatic (as for example in `glm()`).