

# Negative Binomial

## Parametrisation

The negative Binomial distribution is

$$\text{Prob}(y) = \frac{\Gamma(y+n)}{\Gamma(n)\Gamma(y+1)} p^n (1-p)^y$$

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

$n$ : number of successful trials (*size*), or dispersion parameter. Must be strictly positive, need not be integer.

$p$ : probability of success in each trial.

## Link-function

The mean and variance of  $y$  are given as

$$\mu = n \frac{1-p}{p} \quad \text{and} \quad \sigma^2 = \mu \left(1 + \frac{\mu}{n}\right)$$

and the mean is linked to the linear predictor by

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

where the hyperparameter  $n$  (*size*) plays the role of an dispersion parameter.  $E$  represents known constant and  $\log(E)$  is the offset of  $\eta$ .

## Hyperparameters

The default parameterization (**variant=0**) is that the dispersion parameter  $n$  (*size*) is represented as

$$n = \exp(\theta)$$

and the prior is defined on  $\theta$ .

If **variant=1**, then dispersion parameter  $n$  (*size*) is represented as

$$n = E \exp(\theta)$$

and the prior is defined on  $\theta$ .

If **variant=2**, then dispersion parameter  $n$  (*size*) is represented as

$$n = S \exp(\theta)$$

where  $S$  is **scale**, and the prior is defined on  $\theta$ .

## Specification

- family = **nbinomial**
- Required arguments:  $y$  and  $E$  (default  $E = 1$ ) and **scale** (default **scale=1**)
- Chose variant with either `control.family = list(variant=0)` (default) or `control.family = list(variant=1)` or `control.family = list(variant=2)`.

## Hyperparameter specification and default values

**doc** The negBinomial likelihood

**hyper**

**theta**

**hyperid** 63001  
**name** size  
**short.name** size  
**initial** 2.30258509299405  
**fixed** FALSE  
**prior** pc.mgamma  
**param** 7  
**to.theta** function(x) log(x)  
**from.theta** function(x) exp(x)

**survival** FALSE

**discrete** TRUE

**link** default log logoffset quantile

**pdf** nbinoimial

## Example

In the following example we estimate the parameters in a simulated example with negative binomial responses and assign the hyperparameter  $\theta$  a Gaussian prior with mean 0 and precision 0.01

```
n = 1000
x = rnorm(n, sd = 0.2)
eta = 1 + x
E = runif(n, min = 0, max=10)

mu = E * exp(eta)
size = 3
y = rnbinom(n, size=size, mu=mu)
r = inla(y ~ 1 + x, data = data.frame(y, x, E),
        family = "nbinomial", E=E)

mu = E * exp(eta)
size = E*3
y = rnbinom(n, size=size, mu=mu)
rr = inla(y ~ 1 + x, data = data.frame(y, x, E),
        family = "nbinomial",
        control.family = list(variant = 1),
        E=E)
```

## Notes

As  $n \rightarrow \infty$ , the negative Binomial converges to the Poisson distribution. For numerical reasons, if  $n$  is too large:

$$\frac{\mu}{n} < 10^{-4},$$

then the Poisson limit is used.

The PC-prior is available for `variant=1`, see `inla.doc("pc.gamma")`.

## The `nbinomial2` distribution

The negative Binomial distribution is also available in its “pure form”, as the number of excess experiments to get  $n$  successes with a success in the last experiment,

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

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

$n$ : the (fixed) number of successes before stopping,  $n = 1, 2, \dots$ ,

$p$ : the probability of success in each independent trial.

## Link-function

The probability  $p$  is (by default) linked to the linear predictor  $\eta$  as

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

## Hyperparameters

None.

## Specification

- family = `nbinomial2`
- Required arguments:  $y$  and  $N\text{trials}$

The argument `Ntrials` gives the value of  $n$  for each observation.

## Hyperparameter specification and default values

**doc** The `negBinomial2` likelihood

**hyper**

**survival** FALSE

**discrete** TRUE

**link** default logit loga cauchit probit cloglog loglog

**pdf** `nbinomial`

## Example

```
n = 300
x = rnorm(n, sd = 0.2)
eta = 1 + 1.1*x
p = exp(eta)/(1 + exp(eta))
size = sample(1:5, n, replace=TRUE)
y = rnbino(n, size = size, prob = p)
r = inla(y ~ 1 + x, family = "nbinomial2", Ntrials = size,
        data = data.frame(y, x, size))
summary(r)
```