

# Circular Normal (von Mises distribution)

## Parametrisation

The circular Normal or von Mises distribution, has density

$$f(y) = \frac{1}{2\pi I_0(\kappa s)} \exp(\kappa s \cos(y - \mu)),$$

for continuously responses  $y$  where  $|y - \mu| \leq \pi$  and  $|\mu| \leq \pi$ . Here,

$\mu$  is a measure of location, and

$\kappa$  is a measure of the precision,

$s$  is a fixed scaling,  $s > 0$ , and

$I_0$  is the modified Bessel of first kind and order zero

$$I_0(\tau) = \frac{1}{2\pi} \int_0^{2\pi} e^{\tau \cos \alpha} d\alpha.$$

## Link-function

The “mean” of  $y$  is given as  $\mu$  and the mean is linked to the linear predictor as

$$\mu = 2 \arctan(\eta)$$

(Link function “tan”)

## Hyperparameters

The “precision”  $\kappa$  is represented as

$$\theta = \log \kappa$$

and the prior is defined on  $\theta$ .

## Specification

- family="circularnormal"
- Required arguments:  $y$  and  $s$  (argument `scale`)

The scalings have default value 1.

## Hyperparameter spesification and default values

`doc` The circular Gaussian likelihood

`hyper`

`theta`

`hyperid` 67001

`name` log precision parameter

`short.name` prec

`output.name` Precision parameter for the Circular Normal observations

`output.name.intern` Log precision parameter for the Circular Normal observations

```

    initial 2
    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 tan

pdf circular-normal

status disabled

```

## Example

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

```

ilink = function(x) 2*atan(x)
link = function(x) tan(x/2)

n = 300
z = rnorm(n, sd=0.3)
eta = 1 + z
y.pred = ilink(eta)

## create a simple, almost exact, sampler for the circular normal...
kappa = 5
x = seq(-pi, pi, len = 10000)
d = exp(kappa*cos(x))
dd = cumsum(d)
dd = dd /max(dd)
cn.icdf.func = splinefun(dd, x, method = "monoH.FC")
rcn = function(n) cn.icdf.func(runif(n))

y = y.pred + rcn(n)

formula = y ~ 1 + z
r=inla(formula, data = data.frame(y, z),
       family = "circularnormal", control.inla = list(cmin = -Inf))

```

## Notes

Try to use

```
control.inla=list(cmin = -Inf)
```

to avoid systematic-errors for low precisions.