# Bell

## Parametrisation

The Bell distribution is

$$\text{Prob}(y) = \frac{\lambda^y \exp(1 - \exp(\lambda)) B_y}{y!}$$

for responses $y = 0, 1, 2, \ldots$, where $B_y$ are the Bell-numbers ($B_2 = 2, B_5 = 52, B_8 = 4140$, etc). The expected value is $\lambda \exp(\lambda)$ and the variance is $\lambda(1 + \lambda) \exp(\lambda)$.

## Link-function

The mean is linked to the linear predictor by

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

where $E > 0$ is a known constant.

## Hyperparameters

None.

## Specification

- family = `bell`
- Required arguments: (integer-valued) $y$ and $E$ (default 1).

## Example

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

```
library(VGAM) ## dbell
library(gsl)  ## lambert_W0

dbell <- function(y, theta)
    return (theta^y * exp(1-exp(theta)) * bell(y) / factorial(y))

pbell <- function(y, theta)
    return (sum(dbell(0:y, theta)))

rbell <- function(n, theta) {
    ## brute-force in lack of anything easy available
    stopifnot(length(theta) == 1)
    ymax <- 0
    cdf <- 0
    while(cdf < 0.99999) {
        ymax <- ymax + 10
        cdf <- pbell(ymax, theta)
    }
    y <- 0:ymax
    prob <- dbell(y, theta)
    return (sample(y, n, prob = prob, replace = TRUE))
}

## theta <- 2
## hist(rbell(1000, theta))
```

```
n <- 300
x <- rnorm(n)
eta <- 1 + 0.1 * x
mu <- exp(eta)
y <- numeric(n)
for(i in 1:n) {
    theta <- lambert_W0(mu[i])
    y[i] <- rbell(1, theta)
}

r <- inla(y ~ 1 + x, data = data.frame(y, x), family = "bell")
summary(r)
```

## Notes