

## The z-model

### Parametrization

The z-model is an implementation of the “classical” way to define the “random effect” part of a mixed model, through

$$\eta = \dots + Zz$$

where  $Z$  is a  $n \times m$  matrix and  $z$  a vector of length  $m$  representing zero-mean “random effects”. The z-model is defined as the augmented model

$$\tilde{z} = \begin{pmatrix} v \\ z \end{pmatrix}$$

where  $v \sim \mathcal{N}_n(Zz, \kappa I)$ , where  $\kappa$  is a high fixed precision, and where the precision matrix for  $z$  is  $\tau C$  where  $C > 0$  is a  $m \times m$  (fixed) matrix and  $\tau$  is the precision parameter.

### Hyperparameters

The precision parameter of the z-model is represented as

$$\theta = \log(\tau)$$

and prior is assigned to  $\theta$ . The parameter  $\kappa$  is kept fixed at all times.

### Specification

The z-model is specified inside the `f()` function as

```
f(<whatever>, model="z", Z = <Z>, Cmatrix = <Cmat>, hyper = <hyper>,  
  precision = <precision>)
```

where the required `Z`-matrix argument defines the  $Z$  matrix. The (optional) `Cmatrix` defines the  $C$  matrix and is by default taken to be the diagonal matrix with dimension  $m$ . The `precision` parameter defines the value of  $\kappa$ , and `hyper` the hyperparameter specification for  $\tau$ .

If  $Z$  is a  $n \times m$  matrix then the  $C$  matrix must be  $m \times m$  matrix, and  $\tilde{z}$  has length  $n + m$ . The  $n$  first terms of  $\tilde{z}$  is  $v$  and the last  $m$  terms of  $\tilde{z}$  is  $z$ .

If `constr=TRUE` is given, then this is defined as  $\sum_{i=1}^m z_i = 0$ . If `extraconstr` is given, then it is applied to  $\tilde{z}$ , hence `extraconstr$A` must be a  $k \times (n + m)$  matrix where  $k$  is the number of linear constraints.

### Hyperparameter specification and default values

**hyper**

**theta**

**name** log precision

**short.name** prec

**initial** 4

**fixed** FALSE

**prior** loggamma

**param** 1 5e-05

**to.theta** function(x) log(x)

**from.theta** function(x) exp(x)

```

constr FALSE
nrow.ncol FALSE
augmented FALSE
aug.factor 1
aug.constr
n.div.by
n.required TRUE
set.default.values TRUE
pdf z.pdf
status experimental

```

## Example

```

## An example demonstrating two ways to implement the model  $\eta = Zz$ ,
## where  $z \sim N(0, \tau Q)$ .

```

```

## Simulate data
n = 100
m = 10
Z = matrix(rnorm(n*m), n, m)
rho = 0.8
Qz = toeplitz(rho^(0:(m-1)))
prec.fixed = FALSE ## the precision parameter for z
z = inla.qsample(1, Q=Qz)
eta = Z %*% z
s = 0.1 ## noise stdev
s.fixed = TRUE
y = eta + rnorm(n, sd = s)

```

```

## This is normally not needed at all, but it demonstrate how to set
## the 'high precisions': in the z-model, in the A-part of the linear
## predictor, and in the linear predictor iteself.
precision = exp(15)

```

```

## The first approach use the z-model.
r = inla(y ~ -1 + f(idx, model="z", Z=Z,
                    precision=precision,
                    Cmatrix=Qz,
                    hyper = list(
                        prec = list(
                            initial = 0,
                            fixed = prec.fixed,
                            param = c(1, 1)))),
  data = list(y=y, idx=1:n),
  control.family = list(
    hyper = list(
      prec = list(
        initial = log(1/s^2),
        fixed=s.fixed))),
  control.predictor = list(

```

```

        compute=TRUE,
        precision=precision,
        initial = log(precision)))

## The second one uses the A-matrix
rr = inla(y ~ -1 + f(idx, model="generic",
        precision = precision,
        Cmatrix=Qz,
        hyper = list(
            prec = list(
                initial = 0,
                fixed = prec.fixed,
                param = c(1, 1)))),
        data = list(y=y, idx=1:m),
        control.family = list(
            hyper = list(
                prec = list(
                    initial = log(1/s^2),
                    fixed=s.fixed))),
        control.predictor = list(
            compute=TRUE,
            A=Z,
            precision=precision,
            initial = log(precision)))

## Plot some results
par(mfrow=c(2, 2))
plot(r$summary.linear.predictor$mean[1:n], eta,
     main="z-model: (eta.estimated, eta)")
plot(r$summary.linear.predictor$mean[1:n], eta,
     main="generic-model: (eta.estimated, eta)")
plot(r$internal.marginals.hyperpar[[1]],
     main="Prec.param (both)")
lines(rr$internal.marginals.hyperpar[[1]])

## compare (log) marginal likelihood. recall to add the missing part,
## see inla.doc("generic")
print(r$mlik - (rr$mlik + 0.5*log(det(Qz))))

r = inla.hyperpar(r)
rr = inla.hyperpar(rr)
plot(r$internal.marginals.hyperpar[[1]],
     main="Prec.param (improved, both)")
lines(rr$internal.marginals.hyperpar[[1]])

## compare (log) marginal likelihood. recall to add the missing part,
## see inla.doc("generic")
print(r$mlik - (rr$mlik + 0.5*log(det(Qz))))

```

## Notes

None.