R documentation

of 'marginal.Rd'

March 14, 2011

`marginal` *Functions which operates on marginals*

Description

Density, distribution function, quantile function, random generation, interpolation, expectations and transformations of marginals obtained by `inla` or `inla.hyperpar()`.

Usage

```
inla.dmarginal = function(x, marginal, log = FALSE)
inla.pmarginal = function(q, marginal, normalize = TRUE)
inla.qmarginal = function(p, marginal, len = 1024)
inla.rmarginal = function(n, marginal)
inla.smarginal = function(marginal, log = FALSE, extrapolate = 0.0)
inla.emarginal = function(fun, marginal, ...)
inla.tmarginal = function(fun, marginal, n, h.diff, ...)

These functions computes the density (\code{dmarginal}),
the distribution function (\code{pmarginal}),
the quantile function (\code{qmarginal}),
random generation (\code{rmarginal}),
spline smoothing (\code{smarginal}),
computes expected values (\code{emarginal}),
and transforms the marginal (\code{tmarginal}).
```

Arguments

`marginal`	A marginal object from either `inla` or `inla.hyperpar()`, which is either `list(x=c(), y=c())` with density values `y` at locations `x`, or a `matrix(,n,2)` for which the density values are the second column and the locations in the first column.
`fun`	A (vectorised) function like `function(x) exp(x)` to compute the expectation against, or which define the transformation new = fun(old)
`x`	Evaluation points

q	Quantiles
p	Probabilities
n	The number of observations. If length(n) > 1, the length is taken to be the number required. For inla.marginal.transform, its the number of points to use in the new density.
h.diff	The step-length for the numerical differeniation inside inla.marginal.transform
...	Further arguments to be passed to function which expectation is to be computed.
log	Return density or interpolated density in log-scale?
normalize	Renormalise the density after interpolation?
len	Number of locations used to interpolate the distribution function.

Value

inla.smarginal returns list=c(x=c(), y=c()) of interpolated values do extrapolation using the factor given, whereas the remaining function returns what they say they should do.

Author(s)

Havard Rue <hrue@math.ntnu.no>

See Also

inla, inla.hyperpar

Examples

```
## a simple linear regression example
n = 10
x = rnorm(n)
sd = 0.1
y = 1+x + rnorm(n,sd=sd)
res = inla(y ~ 1 + x, data = data.frame(x,y),
           control.data=list(initial = log(1/sd^2),fixed=TRUE))

## chose a marginal and compare the with the results computed by the
## inla-program
r = res$summary.fixed["x",]
m = res$marginals.fixed$x

## compute the the density for exp(r), version 1
r.exp = inla.tmarginal(exp, m)
## or version 2
r.exp = inla.tmarginal(function(x) exp(x), m)

## to plot the marginal, we use the inla.smarginal, which interpolates (in
## log-scale). Compare with some samples.
plot(inla.smarginal(m), type="l")
s = inla.rmarginal(1000, m)
hist(inla.rmarginal(1000, m), add=TRUE, prob=TRUE)
lines(density(s), lty=2)

m1 = inla.emarginal(function(x) x^1, m)
m2 = inla.emarginal(function(x) x^2, m)
stdev = sqrt(m2 - m1^2)
```

```
q = inla.qmarginal(c(0.025,0.975), m)

## inla-program results
print(r)

## inla.marginal-results (they shouldn't be perfect!)
print(c(mean=m1, sd=stdev, "0.025quant" = q[1], "0.975quant" = q[2]))
```

Index