

Package ‘gfilmm’

December 2, 2020

Type Package

Title Generalized Fiducial Inference for Normal Linear Mixed Models

Version 2.0.1

Description Simulation of the generalized fiducial distribution for normal linear mixed models with interval data. Fiducial inference is somehow similar to Bayesian inference, in the sense that it is based on a distribution that represents the uncertainty about the parameters, like the posterior distribution in Bayesian statistics. It does not require a prior distribution, and it yields results close to frequentist results. Reference: Cisewski and Hanig (2012) <doi:10.1214/12-AOS1030>.

License GPL-3

SystemRequirements C++11

Depends R (>= 3.1.0)

Imports Rcpp (>= 1.0.0), lazyeval, stats, spatstat, utils, forcats, rgr, Matrix, parallel

Suggests kde1d, knitr, rmarkdown, lmerTest, emmeans, testthat, AOV1R, GGally, car

LinkingTo Rcpp, RcppEigen

RoxygenNote 7.1.1

Encoding UTF-8

LazyData true

VignetteBuilder knitr

URL <https://github.com/stla/gfilmm>

BugReports <https://github.com/stla/gfilmm/issues>

NeedsCompilation yes

Author Stéphane Laurent [aut, cre],
Jessi Cisewski [aut, ctb] (author of the original Matlab code,
<<https://orcid.org/0000-0002-9656-2272>>)

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2020-12-02 15:50:02 UTC

R topics documented:

gfiCDF	2
gfiConfInt	3
gfilmm	3
gfilmmPredictive	5
gfiQuantile	5
gfiSummary	6
KM41	7
pHdata	7
Index	8

gfiCDF	<i>Fiducial cumulative distribution function</i>
--------	--

Description

Fiducial cumulative distribution function of a parameter of interest.

Usage

```
gfiCDF(parameter, gfi)
```

Arguments

parameter	a right-sided formula defining the parameter of interest, like <code>~ sigma_error / ^ (Intercept)</code>
gfi	a <code>gfilmm</code> object (output of <code>gfilmm</code> or <code>gfilmmPredictive</code>)

Value

The fiducial cumulative distribution function of the parameter.

Examples

```
h <- 0.01
gfi <- gfilmm(
  ~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2
)
F <- gfiCDF(~ sqrt(sigma_block^2 + sigma_error^2) / ^ (Intercept), gfi)
plot(F, xlim = c(0, 0.3), main = "Coefficient of variation",
     ylab = expression("Pr("<="x)"))
F(0.2)
```

gfiConfInt	<i>Fiducial confidence interval</i>
------------	-------------------------------------

Description

Fiducial confidence interval of a parameter of interest.

Usage

```
gfiConfInt(parameter, gfi, conf = 0.95)
```

Arguments

parameter	a right-sided formula defining the parameter of interest, like <code>~ sigma_error / `(Intercept)`</code>
gfi	a <code>gfilmm</code> object (output of <code>gfilmm</code> or <code>gfilmmPredictive</code>)
conf	confidence level

Value

The fiducial confidence interval of the parameter.

Examples

```
h <- 0.01
gfi <- gfilmm(
  ~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2
)
gfiConfInt(~ sqrt(sigma_block^2 + sigma_error^2) / `(Intercept)` , gfi)
```

gfilmm	<i>Generalized fiducial inference</i>
--------	---------------------------------------

Description

Samples the fiducial distributions.

Usage

```
gfilmm(
  y,
  fixed,
  random,
  data,
  N,
  thresh = N/2,
  long = FALSE,
```

```

    seed = NULL,
    nthreads = parallel::detectCores()
  )

## S3 method for class 'gfilmm'
print(x, ...)

```

Arguments

y	a right-sided formula of the form <code>~ cbind(lower, upper)</code> for the interval data
fixed	a right-sided formula for the fixed effects
random	a right-sided formula for the random effects, or NULL for no random effect
data	the data, a dataframe
N	desired number of simulations
thresh	threshold, default N/2; for experts only
long	logical, whether to use long doubles instead of doubles in the algorithm
seed	the seed for the C++ random numbers generator, a positive integer, or NULL to use a random seed
nthreads	number of threads to run the algorithm with parallelized blocks of code
x	a <code>gfilmm</code> object
...	ignored

Value

A list with two components: a dataframe VERTEX, and a vector WEIGHT. It has class `gfilmm`.

References

J. Cisewski and J.Hannig. *Generalized fiducial inference for normal linear mixed models*. The Annals of Statistics 2012, Vol. 40, No. 4, 2102–2127.

Examples

```

h <- 0.01
gfi <- gfilmm(
  ~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2
)
# fiducial cumulative distribution function of the intercept:
Fintercept <- gfiCDF(~ `(Intercept)`, gfi)
plot(Fintercept, xlim = c(40, 65))
# fiducial confidence interval of the intercept:
gfiConfInt(~ `(Intercept)`, gfi)
# fiducial density function of the intercept:
library(kde1d)
kfit <- kde1d(gfi$VERTEX[["(Intercept)"]], weights = gfi$WEIGHT)
curve(dkde1d(x, kfit), from = 45, to = 65)

```

gfilmmPredictive	<i>Generalized fiducial predictive distributions</i>
------------------	--

Description

Simulations of the generalized fiducial predictive distributions.

Usage

```
gfilmmPredictive(gfi, newdata)
```

Arguments

gfi	a <code>gfilmm</code> object
newdata	dataframe in which to look for variables with which to predict, or NULL if the model is an intercept-only model without random effect

Value

A list with two fields: FPD, a dataframe containing the simulations, and WEIGHT, their weight. This is a `gfilmm` object.

Note

Actually the levels of the random effects given in `newdata` can be different from the original levels. For instance, in the example provided below, we enter `block = c("4", "6")`, but we could also enter `block = c("A", "B")`, even though "A" and "B" are not some levels of the block factor. Both options only mean that the two observations to predict are in two different blocks.

Examples

```
gfi <- gfilmm(
  ~ cbind(yield-0.1, yield+0.1), ~ N, ~ block, npk, 2000, nthreads = 2
)
fpd <- gfilmmPredictive(gfi, data.frame(N = c("0", "1"), block = c("4", "6")))
gfiSummary(fpd)
```

gfiQuantile	<i>Quantiles of a fiducial distribution</i>
-------------	---

Description

Quantiles of the fiducial distribution of a parameter of interest.

Usage

```
gfiQuantile(parameter, gfi, probs)
```

Arguments

parameter a right-sided formula defining the parameter of interest, like `~ sigma_error / ^ (Intercept)`
 gfi a `gfilmm` object (output of `gfilmm` or `gfilmmPredictive`)
 probs numeric vector of probabilities

Value

Numeric vector of quantiles, of the same length as `probs`.

Examples

```
h <- 0.01
gfi <- gfilmm(
  ~ cbind(yield-h, yield+h), ~ 1, ~ block, data = npk, N = 5000, nthreads = 2
)
gfiQuantile(~ sqrt(sigma_block^2 + sigma_error^2), gfi, c(25, 50, 75)/100)
```

`gfiSummary`

Summary of fiducial distributions

Description

Summary of the fiducial distributions.

Usage

```
gfiSummary(gfi, conf = 0.95)
```

Arguments

gfi a `gfilmm` object (output of `gfilmm` or `gfilmmPredictive`)
 conf confidence level

Value

A matrix with summary statistics: means, medians, confidence intervals, and probabilities that the standard deviations equal 0.

Examples

```
data(KM41)
h <- 0.005
gfi <- gfilmm(
  ~ cbind(y-h, y+h), ~ 1, ~ Batch, data = KM41, N = 5000, nthreads = 2
)
gfiSummary(gfi)
```

KM41

Krishnamoorthy & Mathew's example 4.1

Description

The dataset used in Krishnammoorthy & Mathew's example 4.1.

Usage

```
data(KM41)
```

Format

A data frame with 25 rows and 2 columns.

References

Krishnamoorthy and Mathew, Statistical Tolerance Regions, Wiley 2009.

Examples

```
data(KM41)
str(KM41)
table(KM41$Batch)
```

pHdata

pH dataset

Description

A dataset from ?? (I don't remember).

Usage

```
data(pHdata)
```

Format

A data frame with 160 rows and 4 columns. Column SIRE is a factor nested in column DAM.

Examples

```
data(pHdata)
str(pHdata)
table(droplevels(pHdata[pHdata$DAM=="D1", "SIRE"]))
table(droplevels(pHdata[pHdata$DAM=="D2", "SIRE"]))
table(droplevels(pHdata[pHdata$DAM=="D3", "SIRE"]))
```

Index

* data

KM41, [7](#)
pHdata, [7](#)

gfiCDF, [2](#)
gfiConfInt, [3](#)
gfilm, [2](#), [3](#), [3](#), [5](#), [6](#)
gfilmPredictive, [2](#), [3](#), [5](#), [6](#)
gfiQuantile, [5](#)
gfiSummary, [6](#)

KM41, [7](#)

pHdata, [7](#)
print.gfilm(gfilm), [3](#)