

Package: binaryMM (via r-universe)

January 9, 2025

Type Package

Title Flexible Marginalized Models for Binary Correlated Outcomes

Version 0.1.1

Description Estimates marginalized mean and dependence model parameters for correlated binary response data. Dependence model may include transition and/or latent variable terms. Methods are described in: Schildcrout and Heagerty (2007) <[doi:10.1111/j.1541-0420.2006.00680.x](https://doi.org/10.1111/j.1541-0420.2006.00680.x)>, Heagerty (1999) <[doi:10.1111/j.0006-341x.1999.00688.x](https://doi.org/10.1111/j.0006-341x.1999.00688.x)>, Heagerty (2002) <[doi:10.1111/j.0006-341x.2002.00342.x](https://doi.org/10.1111/j.0006-341x.2002.00342.x)>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Imports fastGHQuad, MASS, Rcpp

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Jonathan Schildcrout [aut], Nathaniel Mercaldo [aut], Chiara Di Gravio [cre]

Maintainer Chiara Di Gravio <chiara.di.gravio@vanderbilt.edu>

Date/Publication 2022-10-11 21:52:31 UTC

Additional_repositories <https://cranhaven.r-universe.dev>

Repository <https://cranhaven.r-universe.dev>

RemoteUrl <https://github.com/cranhaven/cranhaven.r-universe.dev>

RemoteRef package/binaryMM

RemoteSha f1c959ba1a8f5dea06161d2d26ddb94f2f121096

RemoteSubdir binaryMM

Contents

anova.MMLong	2
datrand	3
GenBinaryY	3
madras	4
mm	5
MMLongit	7
Index	9

anova.MMLong	<i>Comparing Two Models: ANOVA</i>
--------------	------------------------------------

Description

Compute analysis of variance tables for two fitted model objects

Usage

```
## S3 method for class 'MMLong'
anova(object, ...)
```

Arguments

object	a list with two elements. Element 1 is the first model fit using mm, element 2 is the second model fit using mm
...	additional control variables

Value

ANOVA table

Examples

```
data(datrand)
fit1 <- mm(Y~time*binary, t.formula=~1, data=datrand, id=id, step.max=4, verbose=FALSE)
fit2 <- mm(Y~time*binary, t.formula=~1, lv.formula =~1, data=datrand,
id=id, step.max=4, verbose=FALSE)
anova(fit1,fit2)
```

datrand	<i>Simulated data set</i>
---------	---------------------------

Description

A simulated data set. Data were created using fixed marginal mean parameters ($\beta_0, \beta_1, \beta_2, \beta_3$) = (-1.85, -0.15, 1.00, 0.15) and association parameters (γ, σ) = (1.5, 0.0). These data were created assuming an autocorrelation dependence structure.

Usage

```
datrand
```

Format

A data frame with 24999 rows and 4 variables:

id integer. A patient identifier

Y integer. A binary outcome

time double. A time-varying covariate

binary double. A time-invariant covariate

GenBinaryY	<i>Generate binary response data from a Marginalized Transition and Latent Variable Model</i>
------------	---

Description

Generate binary response data from a Marginalized Transition and Latent Variable Model

Usage

```
GenBinaryY(
  mean.formula,
  lv.formula = NULL,
  t.formula = NULL,
  beta = NULL,
  sigma = NULL,
  gamma = NULL,
  id,
  data,
  q = 10
)
```

Arguments

mean.formula	Right hand side of mean model formula
lv.formula	Latent variable model formula (right hand side only)
t.formula	Transition model formula (right hand side only)
beta	a vector of values for mean.formula
sigma	a vector of values for the latent variable portion of the association model (else NULL)
gamma	a vector of values for the transition portion of the association model (else NULL)
id	a vector of cluster identifiers (it should be the same length nrow(data))
data	a required data frame
q	a scalar to denote the number of quadrature points used for GH numerical integration

Value

The function returns a binary response vector.

Examples

```
set.seed(1)
N      = 100
nclust = sample( seq(10,10), N, replace=TRUE)
id     = rep(seq(N), nclust)
Xe     = rep(rbinom(N,size=1,prob=.5), nclust) # binary exposure
time   = unlist( sapply( as.list(nclust), function(ZZ) seq(ZZ)-1 ) )
data   = data.frame(id, time, Xe)
data   = data[order(data$id, data$time),]
newdata = GenBinaryY(mean.formula=~time*Xe, lv.formula=~1, t.formula=~1,
beta=c(-2.5, 0.25, 0.25, 0.1), sigma=1, gamma=1, id=id, data=data, q=20)
```

madras

Madras Longitudinal Schizophrenia Study: Thought Disorder Subset

Description

madras contains a subset of the data from the Madras Longitudinal Schizophrenia Study, which collected monthly symptom data on 86 schizophrenia patients after their initial hospitalization. The primary question of interest is whether subjects with an older age-at-onset tend to recover more or less quickly, and whether female patients recover more or less quickly. Recovery is measured by a reduction in the presentation of symptoms.

Usage

```
madras
```

Format

A data frame with 922 rows and 5 variables:

id integer. An indicator for thought disorders
thought integer. COLUMN_DESCRIPTION
month integer. Months since hospitalization
gender integer. An indicator for female gender
age double. An indicator for age-at-onset ≥ 20 years

Source

Peter Diggle, Patrick J. Heagerty, Kung-Yee Liang, and Scott L. Zeger. Analysis of longitudinal data. Oxford University Press, 2002.

mm

Fit Marginalized Transition and/or Latent Variable Models

Description

Fit a marginalized transition and/or latent variable models (mTLV) as described by Schildcrout and Heagerty 2007.

Usage

```
mm(  
  mean.formula,  
  lv.formula = NULL,  
  t.formula = NULL,  
  id,  
  data,  
  inits = NULL,  
  weight = NULL,  
  offset = NULL,  
  q = 30,  
  step.max = 1,  
  step.tol = 1e-06,  
  hess.eps = 1e-07,  
  adapt.quad = FALSE,  
  verbose = FALSE,  
  iter.lim = 100,  
  return_args = FALSE  
)
```

Arguments

<code>mean.formula</code>	Mean model formula in which a binary variable is regressed on covariates
<code>lv.formula</code>	Latent variable model formula (right hand side only)
<code>t.formula</code>	Transition model formula (right hand side only)
<code>id</code>	a vector of cluster identifiers (it should be the same length of <code>nrow(data)</code>).
<code>data</code>	a required data frame
<code>inits</code>	an optional list of length 3 containing initial values for marginal mean parameters and all dependence parameters. The format of the list should be: (1) estimates of the mean parameters, (2) estimates of the transition parameters (or NULL if only fitting a mLV model) and (3) estimates of the latent variable parameters (or NULL if only fitting a mT model). If NULL, initial values will be automatically generated.
<code>weight</code>	a vector of sampling weights - if using weighted estimating equations. The vector should be the same length of <code>nrow(data)</code> .
<code>offset</code>	an optional offset
<code>q</code>	a scalar to denote the number of quadrature points used to compute the Gauss-Hermite quadrature rule
<code>step.max</code>	a scalar
<code>step.tol</code>	a scalar
<code>hess.eps</code>	a scalar
<code>adapt.quad</code>	an indicator if adaptive quadrature is to be used. NOT CURRENTLY IMPLEMENTED.
<code>verbose</code>	an indicator if model output should be printed to the screen during maximization (or minimization of negative log-likelihood)
<code>iter.lim</code>	a scalar to denote the maximum iteration limit. Default value is 100.
<code>return_args</code>	indicator to denote if attributes of the output should be printed.

Value

This function returns marginal mean (β) and dependence parameters (α) along with the associated model and empirical covariance matrices

Examples

```
data(datrand)
fit <- mm(Y~time*binary, t.formula=~1, data=datrand, id=id, step.max=4, verbose=FALSE)
```

MMLongit

Function used to fit marginalized models

Description

Main function used to fit marginalized models. See `mm()` for a more user-friendly function and examples

Usage

```
MMLongit(
  params,
  id,
  X,
  Y,
  Xgam,
  Xsig,
  Q,
  weight = rep(1, length(Y)),
  offset = rep(0, length(Y)),
  stepmax = 1,
  steptol = 1e-06,
  hess.eps = 1e-07,
  AdaptiveQuad = FALSE,
  verbose = FALSE,
  iterlim
)
```

Arguments

<code>params</code>	a vector of initial values.
<code>id</code>	a vector of cluster identifiers.
<code>X</code>	a design matrix, including intercept, for the mean formula.
<code>Y</code>	a binary vector
<code>Xgam</code>	a design matrix for the transition formula.
<code>Xsig</code>	a design matrix for the latent variable formula.
<code>Q</code>	a scalar denoting the number of quadrature points.
<code>weight</code>	a vector of sampling weights.
<code>offset</code>	an optional offset term.
<code>stepmax</code>	a scalar
<code>steptol</code>	a scalar
<code>hess.eps</code>	a scalar
<code>AdaptiveQuad</code>	an indicator if adaptive quadrature is to be used. NOT CURRENTLY IMPLEMENTED.

`verbose` an indicator if model output should be printed to the screen during maximization (or minimization of negative log-likelihood). See `print.level` in `nlm`.

`iterlim` a scalar to denote the maximum iteration limit used by `nlm`. Default value is 100

Value

This function returns marginal parameters (beta) and dependence parameters (alpha) along with the associated covariance matrices.

Index

* datasets

datrand, 3

madras, 4

anova.MMLong, 2

datrand, 3

GenBinaryY, 3

madras, 4

mm, 5

MMLongit, 7