

Package ‘jmdl’

October 3, 2018

Type Package

Title Joint Mean-Correlation Regression Approach for Discrete Longitudinal Data

Version 0.3.0

Author Weiping Zhang, Yezhen Li, Chengyong Tang

Maintainer Jane Lee <mote@mail.ustc.edu.cn>

Description Fit joint mean-correlation models for discrete longitudinal data (Tang CY,Zhang W, Leng C, 2017 <doi:10.5705/ss.202016.0435>).

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0)

Imports Formula, methods, minqa, boot, mnormt, MASS, mvtnorm

NeedsCompilation yes

RoxygenNote 6.0.1

Repository CRAN

Date/Publication 2018-10-03 13:00:02 UTC

R topics documented:

angle.plot	2
epilepsy2	2
getJMDL	3
jmdl	4
jmdlMod-class	5
lrt.test	5
modular	6
pbc3	7
show_jmdlMod-method	8
toydata	8
un.angle.plot	9

Index	10
--------------	-----------

 angle.plot

Plot Fitted Results and Model Diagnostics

Description

plot (a) fitted angles; (b) fitted correlations vs time lag; (c) the empirical distribution function vs the fitted distribution function; (d) the empirical correlations vs the fitted correlations, when the discrete longitudinal dataset is balanced.

Usage

```
angle.plot(object, time)
```

Arguments

object	a fitted joint mean correlation model of class "JmdlMod", i.e., typically the result of <code>jmdl()</code> .
time	a vector of observation time points

 epilepsy2

epilepsy2 Data

Description

Data from a clinical trial of 59 epileptics. For a baseline, patients were observed for 8 weeks and the number of seizures recorded. The patients were then randomized to treatment by the drug Progabide (31 patients) or to the placebo group (28 patients). They were observed for four 2-week periods and the number of seizures recorded.

Usage

```
data(epilepsy2)
```

Format

A data frame with 280 rows and 7 variables

Details

- id: identifying number
- treat: 1=treated, 0=not
- seizures: number of seizures
- expind: 0=baseline period, 1=treatment period
- timeadj: weeks of period
- age: in years
- time: in weeks

getJMDL	<i>Extract or Get Generalized Components from a Fitted Joint Mean Correlation Model</i>
---------	---

Description

Extract (or "get") "components" - in a generalized sense - from a fitted joint mean correlation model from an object of class "JmdlMod".

Usage

```
getJMDL(object, name)
```

```
## S3 method for class 'jmdlMod'
getJMDL(object, name = c("m", "Y", "X", "W", "offset",
  "theta", "beta", "gamma", "delta", "loglik", "family", "q", "time", "stdbeta",
  "stdgamma", "stddelta"))
```

Arguments

object	a fitted joint mean correlation model of class "JmdlMod", i.e., typically the result of <code>jmdl()</code> .
name	a character vector specifying the name(s) of the "component". possible values are: "m" a vector of number of measurement for each subject "Y" response matrix "X" model matrix for mean structure "W" model matrix for correlation structure (the lower triangular matrix) "offset" a vector to be added to a linear predictor "theta" parameter estimates of joint mean correlation model "beta" parameter estimates for mean structure model "delta" parameter estimates for mean structure model (for Nbinom model) "gamma" parameter estimates for correlation structure (the lower triangular matrix) "stdbeta" standard error for parameter beta "stddelta" standard error for parameter delta "stdgamma" standard error for parameter gamma "loglik" log-likelihood, except for a constant "family" the marginal distributions of the discrete variables "q" degree of polynomial of the time lag to model the lower triangular matrix "time" a vector of time from the data

Methods (by class)

- `jmdlMod`: Extract or Get Generalized Components from a Fitted Joint Mean Correlation Model

Examples

```
mydat <- toydata
fit <- jmdl(Y|id|time ~ X, data = mydat, q = 2, family = 'Bernoulli')
beta <- getJMDL(fit, "beta")
beta
loglik <- getJMDL(fit, "loglik")
loglik
```

jmdl

Fit Joint Mean-Correlation Models For Discrete Longitudinal Data

Description

Fit a joint mean-correlation model to discrete longitudinal data.

Usage

```
jmdl(formula, data = NULL, q = 2, theta = NULL, W.appendix = NULL,
      offset = NULL, family = c("Bernoulli", "Nbinom", "Poisson"))
```

Arguments

formula	a two-sided linear formula object describing the correlation for both the mean and correlation matrix part of the model, with the response, the corresponding subject id and measurement time on the left of a operator~, divided by vertical bars (" ").
data	data frame containing the variables named in formula.
q	degree of polynomial of the time lag to model the lower triangular matrix.
theta	starting values for the parameters in the model.
W.appendix	appendix array to model time-dependent covariates for the lower triangular matrix.
offset	a term to be added to the linear predictor.
family	the marginal distributions of the discrete variables. choose 'Bernoulli', 'Poisson' or 'Nbinom'.

Examples

```
data(toydata)
mydat <- toydata
fit <- jmdl(Y|id|time ~ X, data = mydat, q = 2, family = 'Bernoulli')
```

jmdlMod-class	<i>Class "jmdlMod" of Fitted Joint Mean-Correlation Models.</i>
---------------	---

Description

Class "jmdlMod" of Fitted Joint Mean-Correlation Models.

Slots

call the matched call
 opt the optimization result returned by optimizeJmdl
 args arguments m, Y, X, W, time
 q degree of polynomial of the time lag to model the lower triangular matrix
 std standard error for parameters
 tval t statistic components list
 p p.value components list
 devcomp the deviance components list
 offset a vector to be added to a linear predictor

lrt.test	<i>Pairwise Likelihood Ratio Statistic Test</i>
----------	---

Description

Conducts a pairwise likelihood ratio test for joint mean-correlation regression.

Usage

```
lrt.test(fit, id)
```

Arguments

fit	a fitted joint mean correlation model of class "JmdlMod", i.e., typically the result of jmdl().
id	the id of parameter to test

modular

*Modular Functions for Joint Mean Correlation Model Fits***Description**

Modular Functions for joint mean correlation model fits

Usage

```
ldFormula(formula, data = NULL, q = 2, theta = NULL, W.appendix = NULL,
  offset = NULL, family = c("Bernoulli", "Nbinom", "Poisson"))
```

```
OptimizeJmdl(m, Y, X, W, time, offset = NULL, theta = NULL, family)
```

```
JmdlMod(opt, args, std, tval, p, q, family, offset, mc)
```

Arguments

formula	a two-sided linear formula object describing the covariates for both the mean and correlation matrix part of the model, with the response, the corresponding subject id and measurement time on the left of a operator~, divided by vertical bars (" ").
data	a data frame containing the variables named in formula.
q	degree of polynomial of the time lag to model the lower triangular matrix.
theta	starting values for the parameters in the model.
W.appendix	appendix array to model time-dependent covariates for the lower triangular matrix.
offset	a term to be added to a linear predictor.
family	the marginal distributions of the discrete variables. choose 'Bernoulli', 'Poisson' or 'Nbinom'.
m	an integer vector of number of measurements for each subject.
Y	a matrix of responses for all subjects.
X	model matrix for mean structure model.
W	model array for the lower triangular matrix.
time	a vector of time from the data.
opt	optimized results returned by OptimizJmdl.
args	arguments returned by ldFormula.
std	standard error for parameters.
tval	t statistic.
p	p.value.
mc	matched call from the calling function.

pbc3

pbc3 Data

Description

Followup of 312 randomised patients with primary biliary cirrhosis, a rare autoimmune liver disease, at Mayo Clinic.

Usage

```
data(pbc3)
```

Format

A data frame with 1390 rows and 15 variables

Details

- id: patients identifier; in total there are 312 patients
- years: number of years between registration and the earlier of death, transplantation, or study analysis time
- status: a factor with levels alive, transplanted and dead
- drug: a factor with levels placebo and D-penicil
- age: at registration in years
- sex: a factor with levels male and female
- year: number of years between enrollment and this visit date, remaining values on the line of data refer to this visit
- ascites: a factor with levels No and Yes
- hepatomegaly: a factor with levels No and Yes
- spiders: a factor with levels No and Yes
- edema: a factor with levels No edema (i.e., no edema and no diuretic therapy for edema), edema no diuretics (i.e., edema present without diuretics, or edema resolved by diuretics), and edema despite diuretics (i.e., edema despite diuretic therapy)
- serBilir: serum bilirubin in mg/dl
- albumin: albumin in gm/dl
- platelets: platelets per cubic ml / 1000
- prothrombin: prothrombin time in seconds

show, jmdlMod-method *Print information for jmdlMod-class*

Description

Print information for jmdlMod-class

Usage

```
## S4 method for signature 'jmdlMod'  
show(object)
```

Arguments

object a fitted joint mean correlation model of class "jmdlMod", i.e., typically the result of jmdl().

toydata *toy Data*

Description

A simulation dataset.

Usage

```
data(toydata)
```

Format

A data frame with 50 rows and 4 variables

Details

- id: identifying number
- Y: response variable: 1, 0
- time: time
- X: independent variable

un.angle.plot *Plot Fitted Results*

Description

plot (a) fitted angles; (b) fitted correlations vs time lag, when the discrete longitudinal dataset is unbalanced.

Usage

`un.angle.plot(object)`

Arguments

`object` a fitted joint mean correlation model of class "JmdlMod", i.e., typically the result of `jmdl()`.

Index

*Topic **datasets**

epilepsy2, [2](#)

pb3, [7](#)

toydata, [8](#)

angle.plot, [2](#)

epilepsy2, [2](#)

getJMDL, [3](#)

jmdl, [4](#)

JmdlMod (modular), [6](#)

jmdlMod-class, [5](#)

ldFormula (modular), [6](#)

lrt.test, [5](#)

modular, [6](#)

OptimizeJmdl (modular), [6](#)

pb3, [7](#)

show, jmdlMod-method, [8](#)

toydata, [8](#)

un.angle.plot, [9](#)