

Package: bild (via r-universe)

September 1, 2024

Type Package

Title A Package for BInary Longitudinal Data

Version 1.2-1

Date 2023-10-18

Author M. Helena Goncalves, M. Salome Cabral and Adelchi Azzalini,
apart from a set of Fortran-77 subroutines written by R.
Piessens and E. de Doncker, belonging to the suite ``Quadpack".

Maintainer M. Helena Goncalves <mhgoncal@ualg.pt>

Description Performs logistic regression for binary longitudinal data,
allowing for serial dependence among observations from a given
individual and a random intercept term. Estimation is via
maximization of the exact likelihood of a suitably defined
model. Missing values and unbalanced data are allowed, with
some restrictions. M. Helena Goncalves et al.(2007) <[DOI:10.18637/jss.v046.i09](https://doi.org/10.18637/jss.v046.i09)>.

Encoding UTF-8

NeedsCompilation yes

License GPL (>= 2)

LazyLoad yes

LazyData yes

Depends R (>= 3.1.0), methods, stats, graphics, grDevices, utils

Repository CRAN

Date/Publication 2023-11-03 14:40:02 UTC

Contents

bild-package	2
airpollution	3
anova-methods	5
bild	6
bild-class	10
bild-internal	11

bildControl	11
bildIntegrate	12
fitted-methods	14
fixeff	14
fixeff-methods	15
getAIC	15
getAIC-methods	16
getcoef	16
getcoef-methods	17
getLogLik	17
getLogLik-methods	18
getvcov	18
getvcov-methods	19
locust	19
model.mat	20
model.mat-methods	21
muscatine	22
plot-methods	23
randeff	25
randeff-methods	25
show-methods	26
summary-methods	26
summary.bild-class	27
vareff	28
vareff-methods	29
Index	30

 bild-package

A Package for Binary Longitudinal Data

Description

Performs logistic regression for binary longitudinal data, allowing for serial dependence among observations from a given individual and a random intercept term. Estimation is via maximization of the exact likelihood of a suitably defined model. Missing values and unbalanced data are allowed, with some restrictions.

Details

This package contains functions to perform the fit of parametric models via likelihood method for binary longitudinal data using "S4" classes and methods as implemented in the methods package.

Acknowledgments

We would like to thank the CRAN team for help in the fine tuning of the Fortran code.

Author(s)

M. Helena Goncalves, M.Salome Cabral and Adelchi Azzalini

References

- Azzalini, A. (1994). Logistic regression for autocorrelated data with application to repeated measures. *Biometrika*, 81, 767-775. Amendment: (1997) vol. 84, 989.
- Goncalves, M. Helena (2002) *Likelihood methods for discrete longitudinal data*. PhD thesis, Faculty of Sciences, University of Lisbon.
- Goncalves, M. Helena and Azzalini, A. (2008). Using Markov chains for marginal modelling of binary longitudinal data in an exact likelihood approach. *Metron*, vol LXVI, 2, 157-181.
- Goncalves MH, Cabral MS and Azzalini A (2012). The R Package bild for the Analysis of Binary Longitudinal Data. *Journal of Statistical Software*, 46(9), 1-17.

See Also

[bild-class](#), [bild](#), [Methods](#)

airpollution

Air Pollution

Description

This example is a subset of data from Six Cities study, a longitudinal study of the health effects of air pollution (Ware, J. H. et al., 1984).

Usage

```
data(airpollution)
```

Format

A data frame with 128 observations on the following 5 variables.

- `id` identifies de number of the individual profile. This vector contains observations of 32 individual profiles.
- `wheeze` a numeric vector that identify the wheezing status (1="yes", 0="no") of a child at each occasion.
- `age` a numeric vector corresponding to the age in years since the child's 9th birthday.
- `smoking` a factor that identify if the mother smoke (1="smoke", 0="no smoke").
- `counts` a numeric vector corresponding to the replications of each individual profile.

Details

The data set presented by Fitzmaurice and Laird (1993) contains complete records on 537 children from Steubenville, Ohio, each woman was examined annually at ages 7 through 10. The repeated binary response is the wheezing status (1="yes", 0="no") of a child at each occasion. Although mother's smoking status could vary with time, it was determined in the first interview and was treated as a time-independent covariate. Maternal smoking was categorized as 1 if the mother smoked regularly and 0 otherwise.

Source

Fitzmaurice, G. M. and Laird, N. M. (1993). A Likelihood-Based Method for analyzing Longitudinal Binary Response. *Biometrika*, 80, 141-51.

References

Ware, J. H., Dockery, D. W., Spiro, A. III, Speizer, F. E. and Ferris, B. G., Jr. (1984). Passive smoking, gas cooking and respiratory health in children living in six cities. *Am. Rev. Respir. dis.*, 129, 366-74.

Examples

```
str(airpollution)

##### dependence="MC2"
air2 <- bild(wheeze~age+smoking, data=airpollution, time="age",
            aggregate=smoking, dependence="MC2")

summary(air2)
getAIC(air2)
getLogLik(air2)
plot(air2)

##### dependence="MC2R"

air2r <- bild(wheeze~age+smoking, data=airpollution, time="age",
             aggregate=smoking, dependence="MC2R")

summary(air2r)
getAIC(air2r)
getLogLik(air2r)
plot(air2r)

plot(air2r, which=6, subSET=smoking=="0", main="smoking==0", ident=TRUE)
```

Description

Compute an analysis deviance table for two fitted model objects.

Usage

```
## S4 method for signature 'bild'  
anova(object, ..., test = TRUE, correct = FALSE)
```

Arguments

<code>object</code>	an object of class <code>bild</code> .
<code>...</code>	an object of class <code>bild</code> .
<code>test</code>	an optional logical value controlling whether likelihood ratio tests should be used to compare the fitted models represented by <code>object</code> and by <code>y</code> . The default is <code>TRUE</code> .
<code>correct</code>	an optional logical value controlling whether the p-value of the likelihood ratio test must be corrected. The default is <code>FALSE</code> .

Details

`correct = TRUE` is used to test the presence of a random intercept term and the solution proposed by Self and Liang (1987) is adopted only to the p-value.

Warning

The comparison between two models by `anova` will only be valid if they are fitted to the same dataset.

Methods

`signature(object = "ANY")`: Generic function.
`signature(object="bild")`: Anova for `bild` object.

References

Self, Steven G. and Liang, Kung-Yee (1987). Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association*, 82, 605-610.

Examples

```
##### data = locust

loc1 <- bild(move~(time+I(time^2))*feed*sex, data=locust, dependence="MC1")

loc2 <- bild(move~(time+I(time^2))*feed, data=locust, dependence="MC1")

anova(loc1,loc2)

loc3 <- bild(move~(time+I(time^2))*feed, data=locust, dependence="MC2")

anova(loc3,loc2)

##### data= muscatine

# we decompose the time effect in orthogonal components
muscatine$time1 <- c(-1, 0, 1)
muscatine$time2 <- c(1, -2, 1)

musc1 <- bild(obese~time1, data=muscatine, time="time1",
             dependence="MC1")

musc1r <- bild(obese~time1, data=muscatine, time="time1",
             dependence="MC1R")

anova(musc1, musc1r, correct=TRUE)
```

bild	<i>Fit of Parametric Models for Binary Longitudinal Data via Likelihood Method</i>
------	--

Description

Performs the fit of parametric models via likelihood method. Serial dependence and random intercept are allowed according to the stochastic model chosen. Missing values and unbalanced data are automatically accounted for computing the likelihood function.

Usage

```
bild(formula = formula(data), data, time, id, subSET,
     aggregate = FALSE, start = NULL, trace = FALSE,
     dependence="ind", method = "BFGS",
     control = bildControl(), integrate = bildIntegrate())
```

Arguments

formula a description of the model to be fitted of the form response~predictors

<code>data</code>	a data frame containing the variables in the formula. NA values are allowed. If data is missing, an error message is produced. See "Details".
<code>time</code>	a string that matches the name of the time variable in <code>data</code> . By default, the program expects a variable named <code>time</code> to be present in the <code>data.frame</code> , otherwise the name of the variable playing the role of time must be declared by assigning <code>time</code> here.
<code>id</code>	a string that matches the name of the id variable in <code>data</code> . By default, the program expects a variable named <code>id</code> to be present in the <code>data.frame</code> , otherwise the name of the variable playing the role of id must be declared by assigning <code>id</code> here.
<code>subSET</code>	an optional expression indicating the subset of the rows of data that should be used in the fit. All observations are included by default.
<code>aggregate</code>	a string that permits the user identify the factor to be used in plot-methods .
<code>start</code>	a vector of initial values for the nuisance parameters of the likelihood. The dimension of the vector is according to the structure of the dependence model.
<code>trace</code>	logical flag: if TRUE, details of the nonlinear optimization are printed. By default the flag is set to FALSE.
<code>dependence</code>	expression stating which dependence structure should be used in the fit. The default is "ind". According to the stochastic model chosen serial dependence and random effects are allowed. There are six options: "ind" (independence), "MC1" (first order Markov Chain), "MC2" (second order Markov Chain), "indR" (independence with random intercept), "MC1R" (first order Markov Chain with random intercept) or "MC2R" (second order Markov Chain with random intercept).
<code>method</code>	The method used in the optimization process: "BFGS", "CG", "L-BFGS-B" and "SANN". The default is "BFGS". See optim for details.
<code>control</code>	a list of algorithmic constants for the optimizer <code>optim</code> . See R documentation of <code>optim.control</code> for details and possible control options. By default, <code>bild</code> sets the maximum number of iterations (<code>maxit</code>) equal to 100, the absolute convergence tolerance (<code>abstol</code>) and the relative convergence tolerance (<code>rel.tol</code>) equal to 1e-6 and uses the optim standard default values for the remaining options.
<code>integrate</code>	a list of algorithmic constants for the computation of a definite integral using a Fortran-77 subroutine. See "Details".

Details

`data` are contained in a `data.frame`. Each element of the `data` argument must be identifiable by a name. The simplest situation occurs when all subjects are observed at the same time points. The response variable represent the individual profiles of each subject, it is expected a variable in the `data.frame` that identifies the correspondence of each component of the response variable to the subject that it belongs, by default is named `id` variable. It is expected a variable named `time` to be present in the `data.frame`. If the time component has been given a different name, this should be declared. The time variable should identify the time points that each individual profile has been observed.

When it is expected that all subjects in one experiment to be observed at the same time points, but in practice some of the subjects were not observed in some of the scheduled occasions, NA values can then be inserted in the response variable. If a response profile is replicated several times, a variable called `counts` must be created accordingly. This vector is used for weighting the response profile indicating for each individual profile the number of times that is replicated. The vector `counts` must repeat the number of the observed replications of each individual profile as many times as the number of observed time points for the correspondent profile. The program expect such vector to be named `counts`. If each profile has been observed only once, the construction of the vector `counts` is not required.

`subSET` is an optional expression indicating the subset of data that should be used in the fit. This is a logical statement of the type `variable 1 == "a" & variable 2 > x` which identifies the observations to be selected. All observations are included by default.

For the models with random intercept `indR`, `MC1R` and `MC2R`, `bild` compute integrals based on a Fortran-77 subroutine package `QUADPACK`. For some data sets, when the dependence structure has a random intercept term, the user could have the need to do a specification of the `integrate` argument list changing the integration limits in the `bildIntegrate` function. The `bildIntegrate` is an auxiliary function for controlling `bild` fitting. See the example of `locust` data.

Value

An object of class `bild`.

Background

Assume that each subject of a given set has been observed at number of successive time points. For each subject and for each time point, a binary response variable, taking value 0 and 1, and a set of covariates are recorded. The underlying methodology builds a logistic regression model for the probability that the response variable takes value 1 as a function of the covariates, taking into account that successive observations from the same individual cannot be assumed to be independent.

The basic model for serial dependence is of Markovian type of the first order (denoted `MC1` here), suitably constructed so that the logistic regression parameters maintain the same meaning as in ordinary logistic regression for independent observations. The serial dependence parameter is the logarithm of the odds-ratio between probabilities of adjacent observations, which is assumed to be constant for all adjacent pairs, and it is denoted here `log.psi1`.

An extension of this formulation allows a Markovian dependence of the second order, denoted `MC2` here. In this case there are two parameters which regulate serial dependence: `log.psi1` as before and `log.psi2` which is the analogous quantity for observations which are two time units apart, conditionally on the intermediate value.

Individual random effects can be incorporated in the form of a random intercept term of the linear predictor of the logistic regression, assuming a normal distribution of mean 0 and variance σ^2 , parameterized as $\omega = \log(\sigma^2)$. The combination of serial Markov dependence with a random intercept corresponds here to the dependence structures `MC1R` and `MC2R`. The combination of an independence structure with a random intercept is also allowed setting the dependence structure to `indR`.

Original sources of the above formulation are given by Azzalini (1994), as for the first order Markov dependence, and by Goncalves (2002) and Goncalves and Azzalini (2008) for the its extensions.

Author(s)

M. Helena Goncalves, M. Salome Cabral and Adelchi Azzalini

References

Azzalini, A. (1994). Logistic regression for autocorrelated data with application to repeated measures. *Biometrika*, 81, 767-775. Amendment: (1997) vol. 84, 989.

Goncalves, M. Helena (2002). *Likelihood methods for discrete longitudinal data*. PhD thesis, Faculty of Sciences, University of Lisbon.

Goncalves, M. Helena and Azzalini, A. (2008). Using Markov chains for marginal modelling of binary longitudinal data in an exact likelihood approach. *Metron*, vol LXVI, 2, 157-181.

Goncalves, M. Helena and Cabral, M. Salome and Azzalini, Adelchi (2012). The R Package bild for the Analysis of Binary Longitudinal Data. *Journal of Statistical Software*, 46(9), 1-17.

See Also

[bild-class](#), [bildControl](#), [bildIntegrate](#), [optim](#)

Examples

```
## Are the examples used in respective dataset files

##### data= airpollution, dependence="MC2R"
str(airpollution)

air2r <- bild(wheeze~age+smoking, data=airpollution, trace=TRUE,
             time="age", aggregate=smoking, dependence="MC2R")

summary(air2r)
getAIC(air2r)
getLogLik(air2r)
plot(air2r)

##### data=muscatine, dependence="MC2"
str(muscatine)

# we decompose the time effect in orthogonal components
muscatine$time1 <- c(-1, 0, 1)
muscatine$time2 <- c(1, -2, 1)

musc2 <- bild(obese~(time1+time2)*sex, data=muscatine,
             time="time1", aggregate=sex, trace=TRUE, dependence="MC2")

summary(musc2)
getAIC(musc2)
getLogLik(musc2)
```

 bild-class

 Class "bild" for Results of a Maximum Likelihood Estimation

Description

This class encapsulates results of a maximum likelihood procedure.

Objects from the Class

Objects can be created by calls of the form `new("bild", ...)`, but most often as the result of a call to `build`.

Slots

`coefficients`: Object of class "matrix". Estimated parameters.

`se`: Object of class "matrix". Standard errors of estimated parameters.

`covariance`: Object of class "matrix". Covariance of estimated parameters.

`correlation`: Object of class "matrix". Correlation of estimated parameters.

`log.likelihood`: Object of class "numeric". The value of the log likelihood.

`message`: Object of class "integer". A character string giving any additional information returned by the optimizer, or NULL. See `optim` for details.

`n.cases`: Object of class "numeric". Number of individual profiles used in the optimization procedure.

`ni.cases`: Object of class "numeric". Number of individual profiles in the dataset.

`aic`: Object of class "numeric". The Akaike information criterion for a fitted model object.

`residuals`: Object of class "numeric". The residuals of estimated parameters.

`s.residuals`: Object of class "numeric". The residuals of estimated parameters summed over the individual profile.

`ind.probability`: Object of class "numeric". The transitions probabilities.

`prob.matrix`: Object of class "matrix". The matrix of transitions probabilities.

`Fitted`: Object of class "numeric". The fitted values for the estimated parameters.

`bi.estimate`: Object of class "matrix". The estimated values for the individual random effects.

`Fitted.av`: Object of class "numeric".

`Time`: Object of class "numeric". Vector of time points.

`model.matrix`: Object of class "matrix". The model matrix.

`y.matrix`: Object of class "matrix". The matrix of response values.

`subset.data`: Object of class "data.frame". The data subset if considered.

`y.av`: Object of class "numeric". The average of the response value over an individual profile.

`f.value`: Object of class "factor". Indicates the aggregation factor if present.

`call`: Object of class "language". The call to "build".

Methods

- anova** signature(object="bild"): Display anova table.
- plot** signature(x="bild", y="missing"): Plots six type of plots.
- show** signature(object="bild"): Display object briefly.
- summary** signature(object="bild"): Generate object summary.
- getAIC** signature(object="bild"): Returns a numeric value corresponding to the AIC of the fitted model.
- getLogLik** signature(object="bild"): Returns a numeric value corresponding to the log-Likelihood of the fitted model.
- fitted** signature(object="bild"): The fitted values of a fitted model.
- fixeff** signature(object="bild"): The values corresponding to the fixed effects of a fitted model.
- getcoef** signature(object="bild"): The values corresponding to the coefficient estimates of the fitted model.
- getvcov** signature(object="bild"): The variance-covariance matrix of the fitted model.
- model.mat** signature(object="bild"): The fixed effects model matrix of the fitted model.
- randeff** signature(object="bild"): A data frame corresponding to the conditional random effects of the fitted model.
- vareff** signature(object="bild"): Numeric value corresponding to the estimated random effect variance of the fitted model.

 bild-internal

Internal functions

Description

Functions for internal usage only.

 bildControl

Auxiliary for Controlling "bild" Fitting

Description

Auxiliary function as user interface for bild fitting

Usage

```
bildControl(maxit = 100, abstol = 1e-006, reltol = 1e-006)
```

Arguments

maxit	maximum number of iterations.
abstol	absolute convergence tolerance.
reltol	relative convergence tolerance.

Details

See R documentation of [optim](#) for details of standard default values for the remaining options not considered in bildControl.

Value

A list with the arguments as components.

See Also

[bild-class](#), [optim](#)

bildIntegrate	<i>Auxiliary for Controlling "bild" Fitting</i>
---------------	---

Description

Auxiliary function as user interface for bild fitting

Usage

```
bildIntegrate(li=-4,ls=4, epsabs=.Machine$double.eps^.25,
epsrel=.Machine$double.eps^.25,limit=100,key=6,lig=-4,lsg=4)
```

Arguments

li	lower limit of integration for the log-likelihood.
ls	upper limit of integration for the log-likelihood.
epsabs	absolute accuracy requested.
epsrel	relative accuracy requested.
key	integer from 1 to 6 for choice of local integration rule for number of Gauss-Kronrod quadrature points. A gauss-kronrod pair is used with: 7 - 15 points if key = 1, 10 - 21 points if key = 2, 15 - 31 points if key = 3, 20 - 41 points if key = 4, 25 - 51 points if key = 5 and 30 - 61 points if key = 6.

limit	integer that gives an upperbound on the number of subintervals in the partition of (li,ls), limit.ge.1.
lig	lower limit of integration for the gradient.
lsg	upper limit of integration for the gradient.

Details

bildIntegrate returns a list of constants that are used to compute integrals based on a Fortran-77 subroutine dqage from a Fortran-77 subroutine package QUADPACK for the numerical computation of definite one-dimensional integrals. The subroutine dqage is a simple globally adaptive integrator in which it is possible to choose between 6 pairs of Gauss-Kronrod quadrature formulae for the rule evaluation component. The source code dqage was modified and re-named dqager, the change was the introduction of an extra variable that allow, in our Fortran-77 subroutines when have a call to dqager, to control for which parameter the integral is computed.

For given values of li and ls, the above-described numerical integration is performed over the interval $(li*\sigma, ls*\sigma)$, where $\sigma = \exp(\omega)/2$ is associated to the current parameter value ω examined by the optim function. In some cases, this integration may generate an error, and the user must suitably adjust the values of li and ls. In case different choices of these quantities all lead to a successful run, it is recommended to retain the one with largest value of the log-likelihood. Integration of the gradient is regulated similarly by lig and lsg.

For datasets where the individual profiles have a high number of observed time points (say, more than 30), use bildIntegrate function to set the integration limits for the likelihood and for the gradient to small values than the default ones, see the example of [locust](#) data.

If fitting procedure is complete but when computing the information matrix some NaNs are produced, the change in bildIntegrate function of the default values for the gradient integration limits (lig and lsg) might solve this problem.

Value

A list with the arguments as components.

See Also

[bild-class](#)

Examples

```
## It takes a very long time to run

#### data=locust, dependence="MC2R"
str(locust)

Integ <- bildIntegrate(li=-2.5,ls=2.5, lig=-2.5, lsg=2.5)
locust2r_feed1 <- bild(move~(time+I(time^2))*sex, data=locust,
  trace=TRUE, subSET=feed=="1", aggregate=sex, dependence="MC2R",
  integrate=Integ)

summary(locust2r_feed1)
getAIC(locust2r_feed1)
```

```
getLogLik(locust2r_feed1)
plot(locust2r_feed1)
```

fitted-methods	<i>Methods for function fitted</i>
----------------	------------------------------------

Description

Methods for function `fitted` extracting fitted values of a fitted model object from class `build`.

Usage

```
## S4 method for signature 'build'
fitted(object)
```

Arguments

`object` an object of class `build`.

Methods

`signature(object="build")`: fitted for `build` object.

fixeff	<i>Extract fixed effects estimates</i>
--------	--

Description

Methods for function `fixeff` extracting fixed effects estimates of a fitted model object from class `build`.

Usage

```
fixeff(object)
```

Arguments

`object` an object of class `build`.

Value

Extract fixed effects estimates.

fixeff-methods	<i>Methods for function fixeff</i>
----------------	------------------------------------

Description

Methods for function `fixeff` extracting fixed effects estimates of a fitted model object from class `bild`.

Usage

```
## S4 method for signature 'bild'  
fixeff(object)
```

Arguments

`object` an object of class `bild`.

Methods

`signature(object="bild")`: fixed effects estimates of a fitted model for `bild` object.

getAIC	<i>Extract the Akaike Information Criterion</i>
--------	---

Description

Methods for function `getAIC` extracting the Akaike information criterion for one fitted model object from class `bild`.

Usage

```
getAIC(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns a numeric value corresponding to the AIC of the fitted model.

getAIC-methods

Extract the Akaike Information Criterion

Description

Methods for function `getAIC` extracting the Akaike information criterion for one fitted model object from class `bild`.

Usage

```
## S4 method for signature 'bild'  
getAIC(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns a numeric value corresponding to the AIC of the fitted model.

Methods

getAIC signature(object="bild"):Returns a numeric value corresponding to the AIC of the fitted model.

getcoef

Extract the coefficient estimates

Description

Methods for function `getcoef` extracting the coefficient estimates of the fitted model object from class `bild`.

Usage

```
getcoef(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns the coefficient estimates of the fitted model.

getcoef-methods	<i>Methods for function getcoef</i>
-----------------	-------------------------------------

Description

Methods for function `getcoef` extracting the coefficient estimates of the fitted model object from class `bild`.

Usage

```
## S4 method for signature 'bild'  
getcoef(object)
```

Arguments

`object` an object of class `bild`.

Methods

`signature(object="bild")`: Returns the coefficient estimates of the fitted model `bild` object.

getLogLik	<i>Extract Log-Likelihood</i>
-----------	-------------------------------

Description

Methods for function `getLogLik` extracting the Log-Likelihood for one fitted model object from class `bild`.

Usage

```
getLogLik(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns a numeric value corresponding to the log-Likelihood of the fitted model.

getLogLik-methods *Extract Log-Likelihood*

Description

Methods for function `getLogLik` extracting the Log-Likelihood for one fitted model object from class `bild`.

Usage

```
## S4 method for signature 'bild'  
getLogLik(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns a numeric value corresponding to the log-Likelihood of the fitted model.

Methods

`signature(object="bild")`: Returns a numeric value corresponding to the log-Likelihood of the fitted model.

getvcov *Extract variance-covariance matrix*

Description

Extract the variance-covariance matrix of a fitted model object from class `bild`.

Usage

```
getvcov(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns a numeric value corresponding to the variance-covariance matrix.

getvcov-methods	<i>Methods for function getvcov</i>
-----------------	-------------------------------------

Description

Extract the variance-covariance matrix of a fitted model object from class `build`.

Usage

```
## S4 method for signature 'build'
getvcov(object)
```

Arguments

`object` an object of class `build`.

Methods

`signature(object="build")`: Returns a numeric value corresponding to the variance-covariance matrix of the fixed effect estimates of the fitted model `build` object.

locust	<i>Locust</i>
--------	---------------

Description

This data set was presented by MacDonald and Raubenheimer (1995) and analyze the effect of hunger on locomotory behaviour of 24 locust (*Locusta migratoria*) observed at 161 time points. The subjects were divided in two treatment groups ("fed" and "not fed"), and within each of the two groups, the subjects were alternatively "male" and "female". For the purpose of this analysis the categories of the response variable were "moving" and "not moving". During the observation period, the behavior of each of the subjects was registered every thirty seconds.

Usage

```
data(locust)
```

Format

A data frame with 3864 observations on the following 7 variables.

`id` a numeric vector that identifies de number of the individual profile.

`move` a numeric vector representing the response variable.

`sex` a factor with levels 1 for "male" and 0 for "female".

`time` a numeric vector that identifies de number of the time points observed. The time vector considered was obtained dividing (1:161) by 120 (number of observed periods in 1 hour).

`feed` a factor with levels 0 "no" and 1 "yes".

Details

The response variable, move is the binary type coded as 1 for "moving" and 0 for "not moving". The sex covariate was coded as 1 for "male" and 0 for "female". The feed covariate indicating the treatment group, was coded as 1 for "fed" and 0 for "not fed". Azzalini and Chiogna (1997) also have analyze this data set using their S-plus package `rm.tools`.

Source

MacDonald, I. and Raubenheimer, D. (1995). Hidden Markov models and animal behaviour. *Biometrical Journal*, 37, 701-712

References

Azzalini, A. and Chiogna, M. (1997). S-Plus Tools for the Analysis of Repeated Measures Data. *Computational Statistics*, 12, 53-66

Examples

```
str(locust)

#### dependence="MC2"
locust2_feed1 <- bild(move~(time+I(time^2))*sex, data=locust,
  subSET=feed=="1", aggregate=sex, dependence="MC2")

summary(locust2_feed1)

plot(locust2_feed1, which=5, ylab="probability of locomoting",
  main="Feed=1", add.unadjusted=TRUE)

locust2 <- bild(move~(time+I(time^2))*feed, data=locust,
  aggregate=feed, dependence="MC2")

par(mfrow=c(2,2))
plot(locust2, which=1)
plot(locust2, which=2)
plot(locust2, which=3)
plot(locust2, which=4)
par(mfrow=c(1,1))

plot(locust2, which=5, ylab="probability of locomoting",
  add.unadjusted=TRUE)
```

Description

Methods for function `model.mat` extracting the fixed effects model matrix for a fitted model object from class `bild`.

Usage

```
model.mat(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns a numeric value corresponding to the fixed effects model matrix.

`model.mat-methods` *Methods for function `model.mat`*

Description

Methods for function `model.mat` extracting the fixed effects model matrix for a fitted model object from class `bild`.

Usage

```
## S4 method for signature 'bild'  
model.mat(object)
```

Arguments

`object` an object of class `bild`.

Methods

`signature(object="bild")`: Returns the fixed effects model matrix of the fitted model object from class `bild`.

muscatine

Muscatine

Description

This example is a subset of data from the Muscatine Coronary Risk Factor Study, a longitudinal study of coronary risk factors in school children from Muscatine (Iowa, USA).

Usage

```
data(muscatine)
```

Format

A data frame with 156 observations on the following 7 variables.

`id` identifies the number of the individual profile. This vector contains observation of 52 individuals.

`obese` a numeric vector that identifies the obesity status (1="yes", 0="no") of a child at each occasion.

`sex` a factor with levels 1 for "female" and 0 for "male".

`time` a numeric vector (1,2,3) indicating the observed time points.

`counts` a numeric vector indicating the number of times that each profile is replicated.

Details

The data set presented by Fitzmaurice, Laird and Lipsitz (1994) contains records on 1014 children who were 7-9 years old in 1977 and were examined in 1977, 1979 and 1981. Height and weight were measured in each survey year, and those with relative weight greater than 110. The binary response of interest is whether the child is obese (1) or not (0). However, many data records are incomplete, since not all children participate in all the surveys. This data set was also analyzed by Azzalini (1994).

Source

Fitzmaurice, G. M., Laird, N. M. and Lipsitz, S. R. (1994). Analyzing incomplete longitudinal binary responses: a likelihood based approach. *Biometrics*, 38, 602-612.

References

Azzalini, A. (1994). Logistic regression for autocorrelated data with application to repeated measures. *Biometrika*, 81, 767-775.

Examples

```

str(muscatine)

# we decompose the time effect in orthogonal components
muscatine$time1 <- c(-1, 0, 1)
muscatine$time2 <- c(1, -2, 1)

# second order Markov Chain without random effects
musc2 <- bild(obese~(time1+time2)*sex, data=muscatine,
             time="time1", aggregate=sex, trace=TRUE, dependence="MC2")

summary(musc2)
getAIC(musc2)
getLogLik(musc2)
fitted(musc2)

```

plot-methods

Methods for Function plot in Package "bild"

Description

Six plots (selectable by which) are currently available: a plot of residuals against fitted values (which=1), a plot of standardized residuals against time (which=2), a plot of the autocorrelation function of the residuals (which=3), a plot of the partial autocorrelation function of the residuals (which=4), a plot for the fitted model (which=5) and a plot for the individual mean profile (which=6). By default, the first five are provided.

Usage

```

## S4 method for signature 'bild,missing'
plot(x,which=c(1:5),ylab=NULL,main=NULL,
     ask=prod(par("mfcol"))<length(which)&&dev.interactive(),
     subSET,add.unadjusted=FALSE,ident=FALSE,
     caption=c("Residuals vs Fitted", "Residuals vs Time",
              "ACF residuals", "PACF residuals",
              "Individual mean profiles"), cex.caption=1)

```

Arguments

x	an object of class <code>bild</code> .
which	if a subset of the plots is required, specify a subset of the numbers 1:6.
ylab	label to some plots (which=5 and 6).
main	title to some plots in addition to the caption (which=5 and 6).
ask	logical expression; if TRUE, the user is asked before each plot.

subSET	logical expression indicating elements to keep in individual mean profile plots: missing values are taken as FALSE. The subSET argument is only for option which=6.
add.unadjusted	logical expression indicating whether or not to add the unadjusted fit for plot in which=5.
ident	logical expression indicating whether or not to add the number of the subject to individual mean profile plots. The ident argument is only for option which=6.
caption	captions to appear above the plots.
cex.caption	controls the size of caption.

Details

The option which=5 provides the parametric fitted model if the dependence structure is "ind" (independence), "MC1" (first order Markov Chain) or "MC2" (second order Markov Chain). When the dependence structure is "indR" (independence with random intercept) or "MC1R" (first order Markov Chain with random intercept) or "MC2R" (second order Markov Chain with random intercept) the parametric adjusted fit is provided and the user can set add.unadjusted=TRUE to provide the unadjusted fitted. The option which=6 is used only if the random intercept is present and provides individual mean profile.

Methods

signature(x="ANY", y="ANY"): Generic function.

signature(x="bild", y="missing"): Plot diagnostics for `bild` object.

Examples

```
## It takes a very long time to run
str(locust)

#### dependence="MC2R"
Integ <- bildIntegrate(li=-2.5,ls=2.5, lig=-2.5, lsg=2.5)
locust2r_feed1 <- bild(move~(time+I(time^2))*sex, data=locust,
  subSET=feed=="1", aggregate=sex, dependence="MC2R",
  integrate=Integ)

summary(locust2r_feed1)

plot(locust2r_feed1, which=5, ylab="probability of locomoting",
  add.unadjusted=TRUE)

plot(locust2r_feed1, which=6, subSET=sex=="1",
  main="sex==1 & Feed=1", ident=TRUE)

locust2r <- bild(move~(time+I(time^2))*feed,data=locust,
  trace=TRUE, aggregate=feed, dependence="MC2R",
  integrate=Integ)
```



```

par(mfrow=c(2,2))
plot(locust2r, which=1)
plot(locust2r, which=2)
plot(locust2r, which=3)
plot(locust2r, which=4)
par(mfrow=c(1,1))

plot(locust2r, which=5, ylab="probability of locomoting",
     main="Feed & Unfeed groups", add.unadjusted=TRUE)

plot(locust2r, which=6, ylab="probability of locomoting",
     main="Fed & Unfed groups", ident=TRUE)

```

randeff

Extract conditional random effects

Description

Methods for function `randeff` extracting conditional random effects of a fitted model object from class `bild`.

Usage

```
randeff(object)
```

Arguments

`object` an object of class `bild`.

Value

Returns a `data.frame` corresponding to the conditional random effects of the fitted model.

randeff-methods

Methods for function randeff

Description

Methods for function `randeff` extracting conditional random effects of a fitted model object from class `bild`.

Usage

```
## S4 method for signature 'bild'
randeff(object)
```

Arguments

object an object of class `bild`.

Methods

`signature(object="bild")`: fitted for `bild` object.

Examples

```
str(airpollution)

##### dependence="indR"
air0R <- bild(wheeze~age+smoking, data=airpollution, time="age", dependence="indR")

randeff(air0R)
```

show-methods

Methods for function show in package "bild"

Description

Show objects of classes `bild` and `summary.bild`.

Methods

`signature(object = "bild")` Print simple summary of a `bild` object, just the call, the number of profiles in the fit, the number of coefficients, the value of the log-likelihood and a message giving additional information returned by the optimizer.

`signature(object = "summary.bild")` Shows call, the number of profiles in the fit, table of coefficients, standard errors and p-values, the log-likelihood, the AIC coefficient, and a message giving additional information returned by the optimizer.

summary-methods

Methods for Function summary in Package "bild"

Description

Summarize objects

Usage

```
## S4 method for signature 'bild'
summary(object, cov=FALSE, cor=FALSE)
```

Arguments

object	an object of class <code>bild</code> .
cov	if set to TRUE prints the matrix of covariances between parameters estimates. The default is FALSE.
cor	if set to TRUE prints the matrix of correlations between parameters estimates. The default is FALSE.

Details

Computes and returns a list of summary statistics of the fitted linear model given a `bild` object, using the components (list elements) "call" and "terms" from its argument, plus depending on the structure of the dependence model chosen, the table for the estimates of coefficients will appear `log.psi1` if the dependence structure of the process corresponds to a first-order Markov chain, or both `log.psi1` and `log.psi2` if the dependence structure of the process corresponds to a second-order Markov chain. `log.psi1` is the log-odds ratio between adjacent observations and `log.psi2` is the logarithm of conditional odds ratio for observations separated by one time point. If the structure of the dependence model chosen includes the random intercept (models "indR", "MC1R" and "MC2R") the estimate of the random effect (ω) will also appear where $\omega = \log \sigma^2$.

Methods

`signature(object = "ANY")`: Generic function.

`signature(object = "bild")`: Prints a summary as an object of class `summary.bild`, containing information about the matched call to `bild`, the number of profiles in the data, the number of profiles used in the fit, the log-likelihood, the AIC, a table with estimates, asymptotic SE, t-values and p-values, the estimated correlation and variance-covariance matrix for the estimated parameters if the user wishes, and a message giving additional information returned by the optimizer.

`summary.bild-class` *Class "summary.bild", Summary of "bild" Objects*

Description

Extract of `bild` object.

Objects from the Class

Objects can be created by calls of the form `new("summary.bild", ...)`, but most often by invoking `summary` on an `bild` object. They contain values meant for printing by `show`.

Slots

coefficients: Object of class "matrix". Estimated parameters.
se: Object of class "matrix". Standard errors of estimated parameters.
covariance: Object of class "matrix". Covariance of estimated parameters.
correlation: Object of class "matrix". Correlation of estimated parameters.
log.likelihood: Object of class "numeric". The value of the log likelihood.
message: Object of class "integer". A character string giving any additional information returned by the optimizer, or NULL. See [optim](#) for details.
n.cases: Object of class "numeric". Number of individual profiles used in the optimization procedure.
ni.cases: Object of class "numeric". Number of individual profiles in the dataset.
aic: Object of class "numeric". The Akaike information criterion for a fitted model object.
call: Object of class "language". The call that generated bild object.

Extends

Class "[bild](#)", directly.

Methods

show signature(object = "summary.bild"): Pretty-prints object.

See Also

[bild](#), [bild-class](#)

 vareff

Extract variance of random effects

Description

Methods for function `vareff` extracting the variance of random effects of a fitted model object.

Usage

```
vareff(object)
```

Arguments

`object` an object of class [bild](#).

Value

Returns the variance estimates of random effects of a fitted model.

vareff-methods	<i>Methods for function vareff</i>
----------------	------------------------------------

Description

Methods for function `vareff` extracting the variance estimates of random effects of a fitted model object from class `bild`.

Usage

```
## S4 method for signature 'bild'  
vareff(object)
```

Arguments

`object` an object of class `bild`.

Methods

`signature(object="bild")`: vareff for `bild` object.

Examples

```
str(airpollution)  
  
##### dependence="indR"  
air0R <- bild(wheeze~age+smoking, data=airpollution, time="age", dependence="indR")  
  
vareff(air0R)
```

Index

- * **class**
 - [bild-class](#), [10](#)
 - [summary.bild-class](#), [27](#)
 - * **datasets**
 - [airpollution](#), [3](#)
 - [locust](#), [19](#)
 - [muscatine](#), [22](#)
 - * **function**
 - [bild](#), [6](#)
 - [bild-internal](#), [11](#)
 - [bildControl](#), [11](#)
 - [bildIntegrate](#), [12](#)
 - [fixeff](#), [14](#)
 - [getAIC](#), [15](#)
 - [getcoef](#), [16](#)
 - [getLogLik](#), [17](#)
 - [getvcov](#), [18](#)
 - [model.mat](#), [20](#)
 - [randeff](#), [25](#)
 - [vareff](#), [28](#)
 - * **methods**
 - [anova-methods](#), [5](#)
 - [fitted-methods](#), [14](#)
 - [fixeff-methods](#), [15](#)
 - [getAIC-methods](#), [16](#)
 - [getcoef-methods](#), [17](#)
 - [getLogLik-methods](#), [18](#)
 - [getvcov-methods](#), [19](#)
 - [model.mat-methods](#), [21](#)
 - [plot-methods](#), [23](#)
 - [randeff-methods](#), [25](#)
 - [show-methods](#), [26](#)
 - [summary-methods](#), [26](#)
 - [vareff-methods](#), [29](#)
 - * **package**
 - [bild-package](#), [2](#)
- [airpollution](#), [3](#)
[anova](#), [bild-method \(anova-methods\)](#), [5](#)
[anova-methods](#), [5](#)
- [bgd2m \(bild-internal\)](#), [11](#)
[bild](#), [3](#), [5](#), [6](#), [8](#), [10](#), [14–19](#), [21](#), [23–29](#)
[bild-class](#), [10](#)
[bild-internal](#), [11](#)
[bild-package](#), [2](#)
[bildControl](#), [9](#), [11](#)
[bildIntegrate](#), [8](#), [9](#), [12](#)
[blik2m \(bild-internal\)](#), [11](#)
- [fitted](#), [bild-method \(fitted-methods\)](#), [14](#)
[fitted-methods](#), [14](#)
[fixeff](#), [14](#)
[fixeff](#), [bild-method \(fixeff-methods\)](#), [15](#)
[fixeff-methods](#), [15](#)
- [getAIC](#), [15](#)
[getAIC](#), [bild-method \(getAIC-methods\)](#), [16](#)
[getAIC-methods](#), [16](#)
[getcoef](#), [16](#)
[getcoef](#), [bild-method \(getcoef-methods\)](#),
[17](#)
[getcoef-methods](#), [17](#)
[getLogLik](#), [17](#)
[getLogLik](#), [bild-method \(getLogLik-methods\)](#), [18](#)
[getLogLik-methods](#), [18](#)
[getvcov](#), [18](#)
[getvcov](#), [bild-method \(getvcov-methods\)](#),
[19](#)
[getvcov-methods](#), [19](#)
- [gint \(bild-internal\)](#), [11](#)
[gint1 \(bild-internal\)](#), [11](#)
[gradlogL.bin0 \(bild-internal\)](#), [11](#)
[gradlogL.bin0I \(bild-internal\)](#), [11](#)
[gradlogL.bin1 \(bild-internal\)](#), [11](#)
[gradlogL.bin1I \(bild-internal\)](#), [11](#)
[gradlogL.bin2 \(bild-internal\)](#), [11](#)
[gradlogL.bin2I \(bild-internal\)](#), [11](#)
- [integ \(bild-internal\)](#), [11](#)

`integ1` (bild-internal), 11

`locust`, 8, 13, 19

`logL.bin0` (bild-internal), 11

`logL.bin0I` (bild-internal), 11

`logL.bin1` (bild-internal), 11

`logL.bin1I` (bild-internal), 11

`logL.bin2` (bild-internal), 11

`logL.bin2I` (bild-internal), 11

`mbgd1` (bild-internal), 11

`mblik1` (bild-internal), 11

`Methods`, 3

`model.mat`, 20

`model.mat`, bild-method
(`model.mat-methods`), 21

`model.mat-methods`, 21

`muscatine`, 22

`num.info` (bild-internal), 11

`num.infoI` (bild-internal), 11

`optim`, 7, 9, 10, 12, 28

`plot`, bild, missing-method
(`plot-methods`), 23

`plot-methods`, 23

`randeff`, 25

`randeff`, bild-method (`randeff-methods`),
25

`randeff-methods`, 25

`show`, bild-method (`show-methods`), 26

`show`, `summary.bild`-method
(`show-methods`), 26

`show-methods`, 26

`summary`, bild-method (`summary-methods`),
26

`summary-methods`, 26

`summary.bild`, 27

`summary.bild-class`, 27

`vareff`, 28

`vareff`, bild-method (`vareff-methods`), 29

`vareff-methods`, 29