Package 'mglm4twin'

September 17, 2024

Type Package

Title Multivariate Generalized linear models for twin data

Version 0.4.0

Date 2024-09-11

Author Wagner Hugo Bonat [aut, cre],
Jacob v. B. Hjelmborg [ctb]

Maintainer Wagner Hugo Bonat <wbonat@ufpr.br>

Description Fitting multivariate generalized linear models for twin and family data.

Depends R (>= 4.4.0)

Suggests testthat, knitr, devtools

Imports stats, Matrix, assertthat, graphics, Rcpp (>= 0.12.16)

License GPL-3 | file LICENSE

LazyData TRUE

URL https://github.com/bonatwagner/mglm4twin

BugReports https://github.com/bonatwagner/mglm4twin/issues

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.3.2

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Contents

anthro 2
aux_summary 4
bmi 4
bpdrds 5
coef.mglm4twin 6
covprod 7

ef_core_pearson . . . 7
ef_correction . . . 8
ef_cross_sensitivity . . . 9
ef_cross_variability . . . 9
ef_pearson . . . 10
ef_quasi_score . . . 11
ef_sandwich . . . 12
ef_sensitivity . . . 13
ef_variability . . . 13
fit_mglm . . . 14
gof . . . 15
mglm4twin . . . 16
mt_anova_mglm . . . 18
mt_build_omega . . . 18
mt_build_sigma . . . 19
mt_compute_gen . . . 20
mt_compute_rho . . . 21
mt_delta_method . . . 21
mt_derivative_V_sqrt_beta . . . 22
mt_emp_variability . . . 22
mt_getInformation . . . 23
mt_initial_values . . . 24
mt_link_function . . . 25
mt_list2vec . . . 26
mt_matrix_linear_predictor . . . 27
mt_struc . . . 28
mt_twin . . . 29
mt_updateBeta . . . 29
mt_updateCov . . . 30
mt_variance_function . . . 31
pAIC . . . 32
pBIC . . . 33
pKLIC . . . 34
plogLik . . . 34
print.mglm4twin . . . 35
psydis . . . 36
summary.mglm4twin . . . 37
t0psqi . . . 37
vcov.mglm4twin . . . 38

Index **40**

anthro *Anthropometric measures (weight and height)*

Description

Anthropometric measures (weight and height) on 861 (327 DZ and 534 MZ) twin-pairs. Furthermore, we explore the flexibility of our proposed model class and model the dispersion. The data set is available as an example in the OpenMx package (Neale et al., 2016). We customize the data set for our needs, so make it available organized for use in the mlm4twin package.

- `weight` - Twin weight.
- `height` - Twin height.
- `age` - Twin age.
- `Group` - Twin zygosity (DZ - dizygotic; MZ - monozygotic).
- `Twin` - Twin code.
- `Twin_pair` - Code of twin within the pair (1 and 2).

Usage

```
data(anthro)
```

Format

a `data.frame` with 1722 records and 6 variables.

Source

Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kirkpatrick, R. M., Estabrook, R., Bates, T. C., Maes, H. H. and Boker, S. M. (2016) OpenMx 2.0: Extended structural equation and statistical modeling. Psychometrika, 81, 535–549.

Bonat, W. H. and Hjelmborg, J. v. B. (2020) Multivariate Generalized Linear Models for Twin and Family data. to appear.

Examples

```
require(mglm4twin)
data(anthro, package="mglm4twin")
anthro$age <- (anthro$age - mean(anthro$age))/sd(anthro$age)
anthro$weight <- (anthro$weight - mean(anthro$weight))/sd(anthro$weight)
anthro$height <- (anthro$height - mean(anthro$height))/sd(anthro$height)
form_Wt <- weight ~ age + Group*Twin_pair
form_Ht <- height ~ age + Group*Twin_pair
biv0 <- list("formE1" = ~ age, "formE2" = ~ age, "formE12" = ~ age,
             "formA1" = ~ age, "formA2" = ~ age, "formA12" = ~ age,
             "formC1" = ~ age, "formC2" = ~ age, "formC12" = ~ age)
Z_biv0 <- mt_twin(N_DZ = 327, N_MZ = 534, n_resp = 2, model = "ACE",
                  formula = biv0, data = anthro)
```

```
control_initial <- list()
control_initial$regression <- list("R1" = c(0.13, 0.10, -0.20, -0.02, 0.037),
                                   "R2" = c(0.23, 0.01, -0.27, -0.11, 0.11))
control_initial$power <- list(c(0), c(0))
control_initial$tau <- c(0.15, 0, 0.12, rep(0,15))
fit_0 <- mglm4twin(linear_pred = c(form_Wt, form_Ht), matrix_pred = Z_biv0,
                   control_initial = control_initial,
                   control_algorithm = list(tuning = 0.5),
                   power_fixed = c(TRUE, TRUE), data = anthro)
```

`aux_summary` *Auxiliary function for computing standard errors*

Description

This function provides alternatives ways to compute standard errors.

Usage

```
aux_summary(object, formula, type = "otimist", data, ...)
```

Arguments

`object`	An object of mglm4twin class.
`formula`	formula specifying regression model for each dispersion component.
`type`	Type of standard errors: otimist, robust1 or robust2.
`data`	Data set.
`...`	additional arguments affecting the summary produced.

Value

Internal function returns standard errors.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

bmi *Body mass index*

Description

It is a fairly common data set from the 'mets' package. The dataset consists of 11188 observations, however, in the 'mglm4twin' package we considered only paired twin-pairs. Thus, we opted to circulate the data in this new form to avoid mistakes. The resulting dataset consists of 4271(2788 DZ and 1483 MZ) twin-pairs.

- `bmi` - Continuous trait (body mass index).
- `age` - Twin age.
- `gender` - Twin gender (male and female).
- `Group` - Twin zygosity (DZ - dizygotic; MZ - monozygotic).
- `Twin` - Code of twin pair.
- `Twin_pair` - Code of twin within the pair (1 and 2).

Usage

```
data(bmi)
```

Format

a `data.frame` with 8542 records and 6 variables.

Source

Holst, K. K. and Scheike, T. H. and Hjelmborg, J. B. (2016). The Liability Threshold Model for Censored twin Data. Computational Statistics and Data Analysis 93, pp. 324-335. doi: 10.1016/j.csda.2015.01.014

Bonat, W. H. and Hjelmborg, J. v. B. (2020) Multivariate Generalized Linear Models for Twin and Family data. to appear.

Examples

```
require(mglm4twin)
data(bmi, package="mglm4twin")
form = bmi ~ Group*Twin_pair
ACE = mt_twin(N_DZ = 5576/2, N_MZ = 2966/2, n_resp = 1, model = "ACE")
fit_ACE <- mglm4twin(linear_pred = c(form), matrix_pred = ACE, data = bmi)
```

bpdrds *Bronchopulmonary dysplasia and respiratory distress syndrome on preterm infants*

Description

We use the dataset analysed by Feng et al. (2009) regarding bronchopulmonary dysplasia (BPD) and respiratory distress syndrome (RDS) on preterm infants. Both diseases are lung related and expected to have a genetic component. The dataset consists of 200 twin-pairs being 137 DZ and 63 MZ. Additionally, we considered the covariates: birth weight (BW), gestation age (GA) and gender (female and male).

- `Twin` - Code of twin pair.
- `gender` - Twin age gender (male and female).
- `GA` - Gestation age.
- `BW` - Birth weight.
- `RDS` - Respiratory distress syndrome (binary).
- `BPD` - Bronchopulmonary dysplasia (binary).
- `Group` - Twin zygosity (DZ - dizygotic; MZ - monozygotic).
- `Twin_pair` - Code of twin within the pair (1 and 2).

Usage

```
data(bmi)
```

Format

a `data.frame` with 400 records and 8 variables.

Source

Feng, R., Zhou, G., Zhang, M. and Zhang, H. (2009) Analysis of twin data using sas. Biometrics, 65, 584–589.

Bonat, W. H. and Hjelmborg, J. v. B. (2020) Multivariate Generalized Linear Models for Twin and Family data. to appear.

Examples

```
require(mglm4twin)
data(bpdrds, package="mglm4twin")
form_BPD <- BPD ~ BW + GA + gender + Group*Twin_pair
form_RDS <- RDS ~ BW + GA + gender + Group*Twin_pair
AE <- mt_twin(N_DZ = 137, N_MZ = 63, n_resp = 2, model = "AE")
fitAE <- mglm4twin(linear_pred = c(form_BPD, form_RDS), matrix_pred = AE,
                   link = c("logit","logit"),
                   variance = c("binomialP","binomialP"), data = bpdrds)
```

coef.mglm4twin *Model Coefficients*

Description

Extract model coefficients for objects of mglm4twin class.

Usage

```
## S3 method for class 'mglm4twin'
coef(object, std.error = FALSE, model, response = NULL, ...)
```

Arguments

object	an object of mglm4twin class.
std.error	logical. If TRUE returns the standard errors for the estimates. Default is FALSE.
model	Type of the fitted model. Options are E, AE, CE, ACE and ADE.
response	For which response regression coefficients are extracted. Default = NULL. It means all estimates are returned including dispersion estimates for all responses.
...	additional arguments affecting the summary produced. Note that there is no extra options for mglm4twin object class.

Value

A data.frame with parameters names, estimates, response variable number and parameters type.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

covprod *Cross variability matrix*

Description

Compute the cross-covariance matrix between covariance and regression parameters. Equation (11) of Bonat and Jorgensen (2016).

Usage

```
covprod(A, res, W)
```

Arguments

A	A matrix.
res	A vector of residuals.
W	A matrix of weights.

Author(s)

Wagner Hugo Bonat

---

ef_core_pearson *Core of the Pearson estimating function.*

Description

Core of the Pearson estimating function.

Usage

```
ef_core_pearson(product, inv_C, res, W)
```

Arguments

product	A matrix.
inv_C	A matrix.
res	A vector of weighted residuals.
W	Matrix of weights.
C	A matrix.

Details

It is an internal function.

Value

A vector.

Author(s)

Wagner Hugo Bonat

ef_correction *Pearson correction term*

Description

Compute the correction term associated with the Pearson estimating function.

Usage

```
ef_correction(D_C, inv_J_beta, D, inv_C)
```

Arguments

D_C	A list of matrices.
inv_J_beta	A matrix. In general it is computed based on the output of the [mglm4twin]{ef_quasi_score}.
D	A matrix. In general it is the output of the mt_link_function.
inv_C	A matrix. In general the output of the mt_build_sigma.

Details

It is an internal function useful inside the fitting algorithm.

Value

A vector with the correction terms to be used on the Pearson estimating function.

Author(s)

Wagner Hugo Bonat

ef_cross_sensitivity *Cross-sensitivity*

Description

Compute the cross-sensitivity matrix between regression and covariance parameters. Equation 10 of Bonat and Jorgensen (2015).

Usage

```
ef_cross_sensitivity(
  Product_cov,
  Product_beta,
  n_beta_effective = length(Product_beta)
)
```

Arguments

Product_cov	A list of matrices.
Product_beta	A list of matrices.
n_beta_effective	Numeric. Effective number of regression parameters.

Value

The cross-sensitivity matrix. Equation (10) of Bonat and Jorgensen (2016).

Author(s)

Wagner Hugo Bonat

---

ef_cross_variability *Compute the cross-variability matrix*

Description

Compute the cross-variability matrix between covariance and regression parameters.

Usage

```
ef_cross_variability(Product_cov, inv_C, res, D)
```

Arguments

Product_cov	A list of matrices.
inv_C	A matrix.
res	A vector.
D	A matrix.

Value

The cross-variability matrix between regression and covariance parameters.

Author(s)

Wagner Hugo Bonat

`ef_pearson` *Pearson estimating function*

Description

Compute the Pearson estimating function its sensitivity and variability matrices.

Usage

```
ef_pearson(
  y_vec,
  mu_vec,
  Cfeatures,
  inv_J_beta = NULL,
  D = NULL,
  correct = FALSE,
  compute_variability = FALSE,
  compute_sensitivity = TRUE,
  W
)
```

Arguments

`y_vec`	A vector.
`mu_vec`	A vector.
`Cfeatures`	A list of matrices.
`inv_J_beta`	A matrix.
`D`	A matrix.
`correct`	Logical.
`compute_variability`	Logical.
`compute_sensitivity`	Logical.
`W`	Matrix of weights.

Details

Compute the Pearson estimating function its sensitivity and variability matrices. For more details see Bonat and Jorgensen (2016) equations 6, 7 and 8.

Value

A list with three components: (i) a vector of quasi-score values, (ii) the sensitivity and (iii) variability matrices associated with the Pearson estimating function.

Author(s)

Wagner Hugo Bonat

`ef_quasi_score` *Quasi-score function*

Description

Compute the quasi-score function, its sensitivy and variability matrix.

Usage

```
ef_quasi_score(D, inv_C, y_vec, mu_vec, W)
```

Arguments

`D`	A matrix. In general the output from `mt_link_function`.
`inv_C`	A matrix. In general the output from `mt_build_sigma`.
`y_vec`	A vector.
`mu_vec`	A vector.
`W`	Matrix of weights.

Value

The quasi-score vector, the Sensivity and variability matrices.

Author(s)

Wagner Hugo Bonat

`ef_sandwich` *Matrix product in sandwich form*

Description

The function `ef_sandwich` is just an auxiliar function to compute product matrix in the sandwich form `bord1 * middle * bord2`. An special case appears when computing the derivative of the covariance matrix with respect to the power parameter. Always the bord1 and bord2 should be diagonal matrix. If it is not true, this product is too slow.

Usage

```
ef_sandwich(middle, bord1, bord2)

ef_sandwich_negative(middle, bord1, bord2)

ef_sandwich_power(middle, bord1, bord2)

ef_sandwich_cholesky(bord1, middle, bord2)

ef_multiply(bord1, bord2)

ef_multiply2(bord1, bord2)
```

Arguments

middle	A matrix.
bord1	A matrix.
bord2	A matrix.

Value

The matrix product `bord1 * middle * bord2`.

Author(s)

Wagner Hugo Bonat

---

ef_sensitivity *Sensitivity matrix*

Description

Compute the sensitivity matrix associated with the Pearson estimating function.

Usage

```
ef_sensitivity(product, W = W)
```

Arguments

product	A list of matrix.
W	weights.

Details

This function implements the equation 7 of Bonat and Jorgensen (2016).

Value

The sensitivity matrix associated with the Pearson estimating function.

Author(s)

Wagner Hugo Bonat and Eduardo Elias Ribeiro Jr

`ef_variability` *Variability matrix*

Description

Compute the variability matrix associated with the Pearson estimating function.

Usage

```
ef_variability(sensitivity, product, inv_C, C, res, W)
```

Arguments

`sensitivity`	A matrix. In general the output from `ef_sensitivity`.
`product`	A list of matrix.
`inv_C`	A matrix. In general the output from `mt_build_sigma`.
`C`	A matrix. In general the output from `mt_build_sigma`.
`res`	A vector. The residuals vector, i.e. (y_vec - mu_vec).
`W`	Matrix of weights.

Details

This function implements the equation 8 of Bonat and Jorgensen (2016).

Value

The variability matrix associated witht the Pearson estimating function.

Author(s)

Wagner Hugo Bonat and Eduardo Elias Ribeiro Jr

fit_mglm *Chaser and Reciprocal Likelihood algorithms*

Description

This function implements the two main algorithms used for fitting multivariate generalized linear models to twin data, i.e. The chaser and the reciprocal likelihood algorithms.

Usage

```
fit_mglm(list_initial, list_link, list_variance,
         list_X, list_Z, list_offset, list_Ntrial, list_power_fixed,
         y_vec, correct, max_iter, tol, method,
         tuning, verbose, weights)
```

Arguments

list_initial	a list of initial values for regression and covariance parameters.
list_link	a list specifying the link function names. Options are: "logit", "probit", "cauchit", "cloglog", "loglog", "identity", "log", "sqrt", "1/mu^2" and "inverse". See mt_link_function for details. Default link = "identity".
list_variance	a list specifying the variance function names. Options are: "constant", "tweedie", "poisson_tweedie", "binomialP" and "binomialPQ". See mt_variance_function for details. Default variance = "constant".
list_X	a list of design matrices. See model.matrix for details.
list_Z	a list of knowm matrices to compose the matrix linear predictor.
list_offset	a list of offset values. Default NULL.
list_Ntrial	a list of number of trials, useful only when analysing binomial data. Default 1.
list_power_fixed	a list of logicals indicating if the power parameters should be estimated or not. Default power_fixed = TRUE.
y_vec	a vector of the stacked response variables.
correct	a logical indicating if the algorithm will use the correction term or not. Default correct = FALSE.
max_iter	maximum number of iterations. Default max_iter = 20.
tol	a numeric specyfing the tolerance. Default tol = 1e-04.
method	a string specyfing the method used to fit the models ("chaser" or "rc"). Default method = "chaser".
tuning	a numeric value in general close to zero for the rc method and close to 1 for the chaser method. This argument control the step-length. Default tuning = 1.
verbose	a logical if TRUE print the values of the covariance parameters used on each iteration. Default verbose = FALSE
weights	Vector of weights for model fitting.

Value

A list with estimated regression and covariance parameters. Details about the estimation procedures as iterations, sensitivity, variability are also provided. In general the users do not need to use this function directly. The `mglm4twin` provides GLM interface for fitting multivariate generalized linear models for twin data.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. and Jorgensen, B. (2016) Multivariate covariance generalized linear models. Journal of Royal Statistical Society - Series C 65:649–675.

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

See Also

`mglm4twin`, `mt_matrix_linear_predictor`, `mt_link_function` and `mt_variance_function`.

---

gof *Measures of Goodness-of-Fit*

Description

Extract the pseudo Gaussian log-likelihood (plogLik), pseudo Akaike Information Criterion (pAIC), pseudo Kullback-Leibler Information Criterion (pKLIC) and pseudo Bayesian Information Criterion (pBIC) for objects of `mglm4twin` class.

Usage

```
gof(object)
```

Arguments

`object`	an object or a list of objects representing a model of `mglm4twin` class.

Value

Returns a data frame containing goodness-of-fit measures.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

Wang, M. (2014). Generalized Estimating Equations in Longitudinal Data Analysis: A Review and Recent Developments. Advances in Statistics, 1(1)1–13.

See Also

`plogLik`, `pAIC`, `pKLIC` and `pBIC`.

`mglm4twin` *Fitting Multivariate Covariance Generalized Linear Models to Twin data*

Description

The function `mglm4twin` is used for fitting multivariate generalized linear models to twin data. The models are specified by a set of lists giving a symbolic description of the linear and matrix linear predictors. The user can choose between a list of link and variance functions. The models are fitted using an estimating function approach, combining the quasi-score function for the regression parameters and the Pearson estimating function for the covariance parameters. For details see Bonat and Jorgensen (2016).

Usage

```
mglm4twin(linear_pred, matrix_pred, link, variance,
       offset, Ntrial, power_fixed, weights, data, control_initial,
       contrasts, control_algorithm)
```

Arguments

`linear_pred`	a list of formula see `formula` for details.
`matrix_pred`	a list of matrices, in general the output of `mt_twin` function.
`link`	a list of link functions names. Options are: `"logit"`, `"probit"`, `"cauchit"`, `"cloglog"`, `"loglog"`, `"identity"`, `"log"`, `"sqrt"`, `"1/mu^2"` and `"inverse"`. See `mt_link_function` for details.
`variance`	a list of variance functions names. Options are: `"constant"`, `"tweedie"`, `"poisson_tweedie"`, `"binomialP"` and `"binomialPQ"`. See `mt_variance_function` for details.
`offset`	a list of offset values if any.
`Ntrial`	a list of number of trials on Bernoulli experiments. It is useful only for `binomialP` and `binomialPQ` variance functions.
`power_fixed`	a list of logicals indicating if the values of the power parameter should be estimated or not.

weights	A list of weights for model fitting. Each element of the list should be a vector of weights of size equals the number of observations. Missing observations should be annotated as NA.
data	a data frame.
control_initial	a list of initial values for the fitting algorithm. If no values are supplied automatic initial values will be provided by the function mt_initial_values.
contrasts	extra arguments to passed to model.matrix.
control_algorithm	a list of arguments to be passed for the fitting algorithm. See fit_mglm for details.

Value

mcglm returns an object of class 'mcglm'.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. and Jorgensen, B. (2016) Multivariate covariance generalized linear models. Journal of Royal Statistical Society - Series C 65:649–675.

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

See Also

fit_mglm, mt_link_function and mt_variance_function.

---

mt_anova_mglm *Wald test for dispersion components*

Description

This function provides Wald test for dispersion components.

Usage

```
mt_anova_mglm(object, formula, data, ...)
```

Arguments

object	An object of mglm4twin class.
formula	formula specifying regression model for each dispersion component.
data	Data set.
...	additional arguments affecting the summary produced.

Value

Internal function returns Wald statistics.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

`mt_build_omega` *Build omega matrix*

Description

This function builds Ω matrix.

Usage

```
mt_build_omega(tau, Z)
```

Arguments

`tau`	A vector
`Z`	A list of matrices in general obtained from mt_twin function.

Value

A list with the Ω matrix and its derivatives with respect to τ.

Author(s)

Wagner Hugo Bonat

`mt_build_sigma` *Build variance-covariance matrix*

Description

This function builds a variance-covariance matrix, based on the variance function and Omega matrix.

Usage

```
mt_build_sigma(
  mu,
  tau,
  power,
  Z,
  Ntrial,
  variance,
  power_fixed,
  inverse = FALSE,
  compute_derivative_beta = TRUE
)
```

Arguments

mu	List with expected values and derivatives. In general the output from mt_link_function.
tau	A numeric vector.
power	A list of numeric or numeric vector. It should be one number for all variance functions except binomialPQ, in that case the argument specifies both p and q. Each slot of the list should specify the power parameter for each response variable.
Z	A list of matrices.
Ntrial	Number of trials for binomial data.
variance	String specifing the variance function: constant, tweedie, poisson_tweedie, binomialP or binomialPQ.
power_fixed	Logical if the power parameter is fixed at initial value (TRUE). In the case power_fixed = FALSE the power parameter will be estimated.
compute_derivative_beta	Logical. Compute or not the derivative with respect to regression parameters.

Value

A list with Σ, Σ^{-1} and the derivative of Σ with respect to the power, tau and beta parameters.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

See Also

mt_link_function, mt_variance_function, mt_build_omega.

mt_compute_gen *Compute genetic measures and their standard errors*

Description

Compute genetic measures and their standard errors using the delta method.

Usage

```
mt_compute_gen(Estimates, vcov, model, n_resp)
```

Arguments

Estimates	Table of estimates, standard errors and parameter names.
vcov	Matrix of variance and covariance.
model	String.
n_resp	Numeric. Number of response variables.

Details

It is an internal function useful in general for summary function associated with Twin models.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

mt_compute_rho *Compute correlation and its standard errors*

Description

Compute correlations and standard errors using the delta method.

Usage

```
mt_compute_rho(Estimates, vcov, component, n_resp)
```

Arguments

Estimates	Table of estimates, standard errors and parameter names.
vcov	Matrix of variance and covariance.
component	Logical indicating for which component the correlation will be computed.
n_resp	Numeric. Number of response variables.

Details

It is an internal function useful in general for summary function associated with Twin models.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

`mt_delta_method` *Delta Method*

Description

Compute standard errors for functions of model parameters using the delta method.

Usage

```
mt_delta_method(fx, point, cov, std = TRUE)
```

Arguments

`fx`	A function (string like formula) of model parameters.
`point`	Vector of parameter estimates.
`cov`	Variance-covariance matrix.
`std`	Logical. If TRUE returns the standard error, otherwise return the new variance-covariance.

Details

It is an internal function useful in general for summary function associated with Twin models.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

mt_derivative_V_sqrt_beta *Derivatives of V^1/2 with respect to beta.*

Description

Compute the derivatives of $V^{1/2}$ matrix with respect to the regression parameters beta.

Usage

```
mt_derivative_V_sqrt_beta(D, D_V_sqrt_mu)
```

Arguments

D	A matrix.
D_V_sqrt_mu	A matrix.

Value

A list of matrices, containg the derivatives of $V^{1/2}$ with respect to the regression parameters.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

mt_emp_variability *Empirical variability matrix*

Description

Compute an empirical version of the variability matrix used to estimate robust standard errors.

Usage

```
mt_emp_variability(object, id)
```

Arguments

object	an object or a list of objects representing a model of mglm4twin class.
id	Index indicating independent unit samples.

Value

Returns a matrix.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

`mt_getInformation` *Getting information about model parameters*

Description

This computes all information required about the number of model parameters.

Usage

```
mt_getInformation(list_initial, list_power_fixed, n_resp)
```

Arguments

`list_initial`	A list of initial values.
`list_power_fixed`	A list of logical specyfing if the power parameters should be estimated or not.
`n_resp`	A number specyfing the nmber of response variables.

Value

The number of β's, τ's, power and correlation parameters.

Author(s)

Wagner Hugo Bonat

mt_initial_values *Automatic Initial Values*

Description

This function provides initial values to be used when fitting multivariate generalized linear models by using the function `fit_mglm`. In general the users do not need to use this function, since it is already employed when setting the argument `control_initial = "automatic"` in the `mglm4twin` function. However, if the users want to change some of the initial values, this function can be useful.

Usage

```
mt_initial_values(linear_pred, matrix_pred, link, variance,
                        offset, Ntrial, contrasts, data)
```

Arguments

`linear_pred`	a list of formula see `formula` for details.
`matrix_pred`	a list of known matrices to be used on the matrix linear predictor. See `mt_matrix_linear_predictor` for details.
`link`	a list of link functions names, see `mglm4twin` for details.
`variance`	a list of variance functions names, see `mglm4twin` for details.
`offset`	a list of offset values if any.
`Ntrial`	a list of the number of trials on Bernoulli experiments. It is useful only for `"binomialP"` and `"binomialPQ"` variance functions.
`contrasts`	list of contrasts to be used in the `model.matrix`.
`data`	data frame.

Details

To obtain initial values for multivariate covariance generalized linear models the function `mt_initial_values` fits a generalized linear model (GLM) using the function `glm` with the specified linear predictor and link function for each response variables considering independent observations. The `family` argument is always specified as `quasi`. The link function depends on the specification of the argument `link`. The variance function is always specified as `"mu"` the only excession appears when using `variance = "constant"` then the family argument in the `glm` function is specified as `quasi(link = link, variance = "constant")`. The estimated value of the dispersion parameter from the `glm` function is used as initial value for the first component of the matrix linear predictor, for all other components the value zero is used. The value of the power parameter is always started at 1. In the cases of multivariate models the correlation between response variables is always started at 0.

Value

Return a list of initial values to be used while fitting in the `mglm4twin` function.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

`mt_link_function` *Link Functions*

Description

The `mt_link_function` is a customized call of the `make.link` function.

Given the name of a link function, it returns a list with two elements. The first element is the inverse of the link function applied on the linear predictor $\mu = g^{-1}(X\beta)$. The second element is the derivative of μ with respect to the regression parameters β. It will be useful when computing the quasi-score function.

Usage

```
mt_link_function(beta, X, offset, link)

mt_logit(beta, X, offset)

mt_probit(beta, X, offset)

mt_cauchit(beta, X, offset)

mt_cloglog(beta, X, offset)

mt_loglog(beta, X, offset)

mt_identity(beta, X, offset)

mt_log(beta, X, offset)

mt_sqrt(beta, X, offset)

mt_invmu2(beta, X, offset)

mt_inverse(beta, X, offset)
```

Arguments

`beta`	a numeric vector of regression parameters.
`X`	a design matrix, see `model.matrix` for details.
`offset`	a numeric vector of offset values. It will be sum up on the linear predictor as a covariate with known regression parameter equals one ($\mu = g^{-1}(X\beta + offset)$). If no offset is present in the model, set offset = NULL.

`link` a string specifying the name of the link function. Options are: `"logit"`, `"probit"`, `"cauchit"`, `"cloglog"`, `"loglog"`, `"identity"`, `"log"`, `"sqrt"`, `"1/mu^2"` and `inverse`. A user defined link function can be used (see Details).

Details

The link function is an important component of the multivariate covariance generalized linear models, since it links the expectation of the response variable with the covariates. Let β be a (p x 1) regression parameter vector and X be an (n x p) design matrix. The expected value of the response variable Y is given by

$$E(Y) = g^{-1}(X\beta),$$

where g is the link function and $\eta = X\beta$ is the linear predictor. Let D be a (n x p) matrix whose entries are given by the derivatives of μ with respect to β. Such a matrix will be required for the fitting algorithm. The function `mt_link_function` returns a list where the first element is μ (n x 1) vector and the second is the D (n x p) matrix. A user defined function can also be used. It must be a function with arguments `beta`, `X` and `offset` (set to `NULL` if non needed). The function must return a length 2 named list with `mu` and `D` elements as a vector and a matrix of proper dimensions.

Value

A list with two elements: mu and D (see Details).

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

See Also

`model.matrix`, `make.link`.

Examples

```
x1 <- seq(-1, 1, l = 5)
X <- model.matrix(~ x1)
mt_link_function(beta = c(1,0.5), X = X,
                 offset = NULL, link = 'log')
mt_link_function(beta = c(1,0.5), X = X,
                 offset = rep(10,5), link = 'identity')
```

---

mt_list2vec *Auxiliar function transforms list to a vector.*

Description

This function takes a list of parameters and tranforms to a vector.

Usage

```
mt_list2vec(list_initial, list_power_fixed)
```

Arguments

`list_initial`	A list specifying initial values.
`list_power_fixed`	A list of logical operators specyfing if the power parameter should be estimated or not.

Details

It is an internal function, in general the users never will use this function. It will be useful, only if the user wants to implement a different variance-covariance matrix.

Value

A vector of model parameters.

Author(s)

Wagner Hugo Bonat

`mt_matrix_linear_predictor` *Matrix Linear Predictor*

Description

Compute the matrix linear predictor. It is an internal function, however, since the concept of matrix linear predictor was proposed recently. I decided let this function visible to the interested reader gets some feeling about how it works.

Usage

```
mt_matrix_linear_predictor(tau, Z)
```

Arguments

`tau`	a numeric vector of dispersion parameters.
`Z`	a list of known matrices.

Details

Given a list with a set of known matrices $(Z_0, ..., Z_D)$ the function `mt_matrix_linear_predictor` returns $U = \tau_0 Z_0 + ... + \tau_D Z_D$.

Value

A matrix.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

Bonat, W. H. and Jorgensen, B. (2016) Multivariate covariance generalized linear models. Journal of Royal Statistical Society - Series C 65:649–675.

Examples

```
require(Matrix)
Z0 <- Diagonal(5, 1)
Z1 <- Matrix(rep(1,5)%*%t(rep(1,5)))
Z <- list(Z0, Z1)
mt_matrix_linear_predictor(tau = c(1,0.8), Z = Z)
```

---

mt_struc *Build structural matrix*

Description

This function builds structural matrices to assist multivariate analysis of twin data. It is an internal function used in the mc_twin function.

Usage

```
mt_struc(n_resp)
```

Arguments

n_resp	Number of response variables.

Value

A list of matrices.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

mt_twin *Build twin matrix linear predictor components*

Description

This function provides the components of the matrix linear predictor suitable for fitting multivariate twin models.

Usage

```
mt_twin(N_DZ, N_MZ, n_resp, model, formula = NULL, data = NULL)
```

Arguments

N_DZ	Number of DZ twin pairs.
N_MZ	Number of MZ twin pairs.
n_resp	Number of response variables.
model	a string specifying the name of the twin model. Options are: "E", "AE", "CE", "ACE" and "ADE" models.
formula	formula specifying regression model for each dispersion component.
data	Data set.

Value

A list of matrices.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

mt_updateBeta *Updated regression parameters*

Description

This function update a list of regression parameters. It will be useful only inside the fitting algorithm.

Usage

```
mt_updateBeta(list_initial, betas, information, n_resp)
```

Arguments

list_initial	A list of initial values.
betas	A vector with actual regression parameters values.
information	A list with information about the number of parameters in the model. In general the output from mt_getInformation.
n_resp	A numeric specyfing the number of response variables.

Value

A list with updated values of the regression parameters.

Author(s)

Wagner Hugo Bonat

---

mt_updateCov *Updated covariance parameters*

Description

This function update a list of covariance parameters. It will be useful only inside the fitting algorithm.

Usage

```
mt_updateCov(list_initial, covariance, list_power_fixed, information, n_resp)
```

Arguments

list_initial	A list of initial values.
covariance	A vector with actual covariance parameters values.
list_power_fixed	A list of logicals indicating if the power parameter should be estimated or not.
information	A list with information about the number of parameters in the model. In general the output from mt_getInformation.
n_resp	A numeric specyfing the number of response variables.

Value

A list with updated values of the covariance parameters.

Author(s)

Wagner Hugo Bonat

mt_variance_function *Variance Functions*

Description

Compute the variance function and its derivatives with respect to regression, dispersion and power parameters.

Usage

```
mt_variance_function(mu, power, Ntrial, variance,
                             derivative_power, derivative_mu)

mt_tweedie(mu, power, Ntrial, derivative_power, derivative_mu)

mt_binomialP(mu, power, Ntrial,
                     derivative_power, derivative_mu)

mt_binomialPQ(mu, power, Ntrial,
                      derivative_power, derivative_mu)

mt_constant(mu, power, Ntrial, derivative_power, derivative_mu)
```

Arguments

mu	a numeric vector. In general the output from mt_link_function.
power	a numeric value (tweedie and binomialP) or a vector (binomialPQ) of the power parameters.
Ntrial	number of trials, useful only when dealing with binomial response variables.
variance	a string specifying the name (constant, tweedie, binomialP or binomialPQ) of the variance function.
derivative_power	logical if compute (TRUE) or not (FALSE) the derivatives with respect to the power parameter.
derivative_mu	logical if compute (TRUE) or not (FALSE) the derivative with respect to the mu parameter.

Details

The function mt_variance_function computes three features related with the variance function. Depending on the logical arguments, the function returns $V^{1/2}$ and its derivatives with respect to the parameters power and mu, respectivelly. The output is a named list, completely informative about what the function has been computed. For example, if derivative_power = TRUE and derivative_mu = TRUE. The output will be a list, with three elements: V_sqrt, D_V_sqrt_power and D_V_sqrt_mu.

Value

A list with from one to four elements depending on the arguments.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. and Jorgensen, B. (2016) Multivariate covariance generalized linear models. Journal of Royal Statistical Society - Series C 65:649–675.

See Also

mt_link_function.

Examples

```
x1 <- seq(-1, 1, l = 5)
X <- model.matrix(~x1)
mu <- mt_link_function(beta = c(1, 0.5), X = X, offset = NULL,
                       link = "logit")
mt_variance_function(mu = mu$mu, power = c(2, 1), Ntrial = 1,
                     variance = "binomialPQ",
                     derivative_power = TRUE, derivative_mu = TRUE)
```

---

pAIC *Pseudo Akaike Information Criterion*

Description

Extract the pseudo Akaike information criterion (pAIC) for objects of mglm4twin class.

Usage

```
pAIC(object, verbose = TRUE)
```

Arguments

object	an object or a list of objects representing a model of mglm4twin class.
verbose	logical. Print or not the pAIC value.

Value

Returns the value of the pseudo Akaike information criterion (pAIC).

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

See Also

gof, plogLik, pKLIC and pBIC.

---

pBIC *Pseudo Bayesian Information Criterion*

Description

Extract the pseudo Bayesian information criterion (pBIC) for objects of mglm4twin class.

Usage

```
pBIC(object, verbose = TRUE)
```

Arguments

object	an object or a list of objects representing a model of mglm4twin class.
verbose	logical. Print or not the pBIC value.

Value

Returns the value of the pseudo Bayesian information criterion (pBIC).

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

See Also

gof, plogLik, pKLIC and pAIC.

pKLIC *Pseudo Kullback-Leibler Information Criterion*

Description

Extract the pseudo Kullback-Leibler information criterion (pKLIC) for objects of `mglm4twin` class.

Usage

```
pKLIC(object, verbose = TRUE)
```

Arguments

`object`	an object or a list of objects representing a model of `mglm4twin` class.
`verbose`	logical. Print or not the pKLIC value.

Value

Returns the value of the pseudo Kullback-Leibler information criterion.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Source

Bonat, W. H. (2018). Multiple Response Variables Regression Models in R: The mcglm Package. Journal of Statistical Software, 84(4):1–30.

See Also

`gof`, `plogLik`, `pAIC`, `pBIC` and `pKLIC`.

plogLik *Gaussian Pseudo-loglikelihood*

Description

Extract the Gaussian pseudo-loglikelihood (plogLik) value for objects of `mglm4twin` class.

Usage

```
plogLik(object, verbose = TRUE)
```

Arguments

object	an object or a list of objects representing a model of mglm4twin class.
verbose	logical. Print or not the plogLik value.

Value

Returns the value of the Gaussian pseudo-loglikelihood.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

---

print.mglm4twin *Print*

Description

The default print method for an object of mglm4twin class.

Usage

```
## S3 method for class 'mglm4twin'
print(x, ...)
```

Arguments

x	fitted model objects of class mglm4twin as produced by mglm4twin().
...	further arguments passed to or from other methods.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

See Also

summary.

psydis *Psychiatric disorders*

Description

Psychiatric disorders in 1030 (440 DZ and 590 MZ) Caucasian female twin-pairs sampled from the Virginia Twin Registry. Lifetime psychiatric illness is a binary trait and was diagnosed using an adapted version of the Structured Clinical Interview for DSM-II-R Diagnosis.

- `y` - Binary trait (disease presence YES - 1; NO - 0).
- `Group` - Twin zygosity (DZ - dizygotic; MZ - monozygotic).
- `Twin` - Code of twin pair.
- `Twin_pair` - Code of twin within the pair (1 and 2).

Usage

```
data(psydis)
```

Format

a `data.frame` with 2060 records and 4 variables.

Source

Neale, M. C. and Maes, H. H. (2004) . Methodology for Genetic Studies of Twins and Families. Tech. rep., Virginia Common wealth University, Department of Psychiatry.

Rabe-Hesketh, S., Skrondal, A. and Gjessing, H. K. (2008) Biometrical modeling of twin and family data using standard mixed model software. Biometrics, 64, 280–288.

Bonat, W. H. and v. B. Hjelmborg, J. (2020) Multivariate Generalized Linear Models for Twin and Family data. to appear.

Examples

```
require(mglm4twin)
data(psydis, package="mglm4twin")
ex1_form <- y ~ 1
ex1_AE <- mt_twin(N_DZ = 440, N_MZ = 590, n_resp = 1, model = "AE")
ex1_AE <- mglm4twin(c(ex1_form), matrix_pred = ex1_AE,
                    link = c("logit"), variance = c("binomialP"),
                   data = psydis)
summary(ex1_AE, model = "AE")
summary(ex1_AE, model = "AE", biometric = TRUE)
```

summary.mglm4twin *Summarizing*

Description

The default summary method for an object of mglm4twin class.

Usage

```
## S3 method for class 'mglm4twin'
summary(object, model, biometric = FALSE, ...)
```

Arguments

object	an object of mglm4twin class.
model	String indicating twin model.
biometric	Logical. If TRUE biometric measures such as heritability, common environment, genetic correlation etc are printed. (Default = FALSE).
...	additional arguments affecting the summary produced. Note the there is no extra options for mglm4twin object class.

Value

Print a mglm4twin object.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

See Also

print.

t0psqi *Sleep's quality*

Description

Data set concerning sleep's quality in a sample of 250 (135 DZ and 116 MZ) Danish twin pairs. The traits are cortisone levels when waking up (T0) and PSQI (Pittsburgh Sleep Quality Index). It is a simulated data set based on the parameter estimates obtained fitting the model to a motivating real data set. The code for the simulation is available in the folder data-raw.

- Twin_pair - Code of twin within the pair (1 and 2).
- Twin_id - Twin code.

- Age - Twin age.
- Type - Twin zygosity (DZ - dizygotic; MZ - monozygotic).
- Gender - Gender (Male and Female).
- Group - Treatment group, it is categorical covariate for composing the linear predictor.
- T0 - Cortisone levels when waking up (continuous trait).
- PSQI - Pittsburgh Sleep Quality Index (bounded trait) divided by 21 (scale maximum).

Usage

```
data(t0psqi)
```

Format

a data.frame with 502 records and 8 variables.

Source

Bonat, W. H. and Hjelmborg, J. v. B. (2020) Multivariate Generalized Linear Models for Twin and Family data. to appear.

Examples

```
require(mglm4twin)
form_T0 <- T0 ~ Age + Gender + Group + Type*Twin_pair
form_PSQI <- PSQI ~ Age + Gender + Group + Type*Twin_pair
AE <- mt_twin(N_DZ = 135, N_MZ = 116, n_resp = 2, model = "AE")
fit_AE <- mglm4twin(linear_pred = c(form_T0,  form_PSQI),
                    matrix_pred = AE,
                    link = c("log","logit"),
                    variance = c("tweedie","binomialP"),
                    control_algorithm = list(tuning = 0.25, max_iter = 100),
                    power_fixed = c(FALSE,FALSE), data = t0psqi)
```

---

vcov.mglm4twin *Variance-Covariance Matrix*

Description

Returns the variance-covariance matrix for an object of mglm4twin class.

Usage

```
## S3 method for class 'mglm4twin'
vcov(object, model, ...)
```

Arguments

`object`	an object of `mglm4twin` class.
`model`	String indicating twin model.
`...`	additional arguments affecting the summary produced. Note that there is no extra options for mcglm object class.

Value

A variance-covariance matrix.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

Index

∗ **datasets**
anthro, 2
bmi, 4
bpdrds, 5
psydis, 36
t0psqi, 37
∗ **internal**
aux_summary, 4
covprod, 7
ef_core_pearson, 7
ef_correction, 8
ef_cross_sensitivity, 9
ef_cross_variability, 9
ef_pearson, 10
ef_quasi_score, 11
ef_sandwich, 12
ef_sensitivity, 13
ef_variability, 13
mt_anova_mglm, 18
mt_build_omega, 18
mt_build_sigma, 19
mt_compute_gen, 20
mt_compute_rho, 21
mt_delta_method, 21
mt_getInformation, 23
mt_list2vec, 26
mt_struc, 28
mt_twin, 29
mt_updateBeta, 29
mt_updateCov, 30

anthro, 2
aux_summary, 4

bmi, 4
bpdrds, 5

coef.mglm4twin, 6
covprod, 7

ef_core_pearson, 7
ef_correction, 8
ef_cross_sensitivity, 9
ef_cross_variability, 9
ef_multiply *(*ef_sandwich*)*, 12
ef_multiply2 *(*ef_sandwich*)*, 12
ef_pearson, 10
ef_quasi_score, 11
ef_sandwich, 12
ef_sandwich_cholesky *(*ef_sandwich*)*, 12
ef_sandwich_negative *(*ef_sandwich*)*, 12
ef_sandwich_power *(*ef_sandwich*)*, 12
ef_sensitivity, 13
ef_variability, 13

fit_mglm, 14, *17*
formula, *17*, *24*

gof, 15

make.link, *25, 26*
mglm4twin, *15*, 16, *24*
model.matrix, *14*, *17*, *24–26*
mt_anova_mglm, 18
mt_binomialP *(*mt_variance_function*)*, 31
mt_binomialPQ *(*mt_variance_function*)*, 31
mt_build_omega, 18, *20*
mt_build_sigma, *8*, *11*, 19
mt_cauchit *(*mt_link_function*)*, 25
mt_cloglog *(*mt_link_function*)*, 25
mt_compute_gen, 20
mt_compute_rho, 21
mt_constant *(*mt_variance_function*)*, 31
mt_delta_method, 21
mt_derivative_V_sqrt_beta, 22
mt_emp_variability, 22
mt_getInformation, 23, *30*
mt_identity *(*mt_link_function*)*, 25
mt_initial_values, *17*, 24
mt_inverse *(*mt_link_function*)*, 25
mt_invmu2 *(*mt_link_function*)*, 25

mt_link_function, *8*, *11*, *14*, *17*, *19*, *20*, 25, *31*, *32*
mt_list2vec, 26
mt_log *(*mt_link_function*)*, 25
mt_logit *(*mt_link_function*)*, 25
mt_loglog *(*mt_link_function*)*, 25
mt_matrix_linear_predictor, *24*, 27
mt_probit *(*mt_link_function*)*, 25
mt_sqrt *(*mt_link_function*)*, 25
mt_struc, 28
mt_tweedie *(*mt_variance_function*)*, 31
mt_twin, 29
mt_updateBeta, 29
mt_updateCov, 30
mt_variance_function, *14*, *17*, *20*, 31

pAIC, 32
pBIC, 33
pKLIC, 34
plogLik, 34
print.mglm4twin, 35
psydis, 36

summary.mglm4twin, 37

t0psqi, 37

vcov.mglm4twin, 38