Package ‘mcglm’

May 8, 2026

Type Package

Title Multivariate Covariance Generalized Linear Models

Version 0.9.0

Date 2026-01-07

Author Wagner Hugo Bonat [aut, cre]

Maintainer Wagner Hugo Bonat <`wbonat@ufpr.br`>

Description Fitting multivariate covariance generalized linear models (McGLMs) to data. McGLM is a general framework for non-normal multivariate data analysis, designed to handle multivariate response variables, along with a wide range of temporal and spatial correlation structures defined in terms of a covariance link function combined with a matrix linear predictor involving known matrices.
The models take non-normality into account in the conventional way by means of a variance function, and the mean structure is modelled by means of a link function and a linear predictor.
The models are fitted using an efficient Newton scoring algorithm based on quasi-likelihood and Pearson estimating functions, using only second-moment assumptions. This provides a unified approach to a wide variety of different types of response variables and covariance structures, including multivariate extensions of repeated measures, time series, longitudinal, spatial and spatio-temporal structures.
The package offers a user-friendly interface for fitting McGLMs similar to the glm() R function.
See Bonat (2018) <doi:10.18637/jss.v084.i04>, for more information and examples.

Depends R (>= 4.5.0)

Suggests testthat, knitr, rmarkdown, MASS, mvtnorm, tweedie, devtools

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

License GPL-3

LazyData TRUE

URL `https://github.com/bonatwagner/mcglm`

BugReports `https://github.com/bonatwagner/mcglm/issues`

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.3.3

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Repository CRAN

Date/Publication 2026-01-08 19:20:07 UTC

Contents

`ahs` *Australian Health Survey Data*

Description

The Australian Health Survey (AHS) was used by Bonat and Jorgensen (2016) as an example of multivariate count regression modeling. The dataset contains five count response variables related to health system usage and nine covariates related to social conditions in Australia for the years 1987-88.

Usage

```
data(ahs)
```

Format

A `data.frame` with 5190 observations and 15 variables:

`sex` Factor with levels `male` and `female`.

`age` Respondent's age in years divided by 100.

`income` Respondent's annual income in Australian dollars divided by 1000.

`levyplus` Factor indicating coverage by private health insurance for private patients in public hospital with doctor of choice (1) or otherwise (0).

`freepoor` Factor indicating government coverage due to low income, recent immigration, or unemployment (1) or otherwise (0).

`freerepa` Factor indicating government coverage due to old-age/disability pension, veteran status, or family of deceased veteran (1) or otherwise (0).

`illnes` Number of illnesses in the past two weeks, capped at 5.

`actdays` Number of days of reduced activity in the past two weeks due to illness or injury.

`hscore` General health questionnaire score (Goldberg's method); higher scores indicate poorer health.

`chcond` Factor with levels: `limited` (chronic condition with activity limitation), `nonlimited` (chronic condition without limitation), `otherwise` (reference level).

`Ndoc` Number of consultations with a doctor or specialist (response variable).

`Nndoc` Number of consultations with health professionals (response variable).

`Nadm` Number of admissions to hospital, psychiatric hospital, nursing, or convalescence home in the past 12 months (response variable).

`Nhosp` Number of nights in a hospital during the most recent admission.

`Nmed` Total number of prescribed and non-prescribed medications used in the past two days.

Source

Deb, P. and Trivedi, P. K. (1997) "Demand for medical care by the elderly: A finite mixture approach." *Journal of Applied Econometrics*, 12(3):313–336.

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

Examples

```
library(mcglm)
data(ahs, package = "mcglm")
form1 <- Ndoc ~ income + age
form2 <- Nndoc ~ income + age
Z0 <- mc_id(ahs)
fit.ahs <- mcglm(linear_pred = c(form1, form2),
                 matrix_pred = list(Z0, Z0),
                 link = c("log", "log"),
                 variance = c("poisson_tweedie", "poisson_tweedie"),
                 data = ahs)
summary(fit.ahs)
```

`anova.mcglm` *Wald Tests for Fixed Effects in mcglm Models*

Description

Performs Wald chi-square tests for assessing the significance of fixed-effect terms in the linear predictors of an `mcglm` model. The tests are conducted separately for each response variable and are particularly useful for joint hypothesis testing of regression coefficients associated with categorical covariates with more than two levels. This function is not intended for model comparison.

Usage

```
## S3 method for class 'mcglm'
anova(object, ..., verbose = TRUE)
```

Arguments

`object`	An object of class `mcglm`, typically the result of a call to `mcglm`.
`...`	Additional arguments. Currently ignored.
`verbose`	Logical indicating whether the Wald test results should be printed to the console. If `FALSE`, the function silently returns the results.

Details

The Wald tests are computed using the observed covariance matrix of the regression parameter estimates. For each response variable, joint tests are performed for sets of parameters corresponding to the same model term, as defined by the design matrix.

Value

A list of data frames, one for each response variable. Each data frame contains the results of Wald chi-square tests for the fixed-effect terms in the corresponding linear predictor, with the following columns:

Covariate Name of the covariate or model term tested.

Chi.Square Value of the Wald chi-square statistic.

Df Degrees of freedom associated with the test.

p.value P-value of the Wald test.

The returned object is invisible and is primarily intended for programmatic use.

Author(s)

Wagner Hugo Bonat, <`wbonat@ufpr.br`>

See Also

`summary.mcglm`, `coef.mcglm`, `vcov.mcglm`

Examples

```
x1 <- seq(-1, 1, length.out = 100)
x2 <- gl(5, 20)
beta <- c(5, 0, -2, -1, 1, 2)
X <- model.matrix(~ x1 + x2)
set.seed(123)
y <- rnorm(100, mean = X %*% beta, sd = 1)
data <- data.frame(y = y, x1 = x1, x2 = x2)
fit <- mcglm(c(y ~ x1 + x2), list(mc_id(data)), data = data)
anova(fit)
```

`coef.mcglm` *Model Coefficients*

Description

Extract regression, dispersion and correlation parameter estimates from objects of class `mcglm`.

Usage

```
## S3 method for class 'mcglm'
coef(
  object,
  std.error = FALSE,
  response = c(NA, 1:length(object$beta_names)),
  type = c("beta", "tau", "power", "correlation"),
  ...
)
```

Arguments

`object`	An object of class `mcglm`.
`std.error`	Logical indicating whether standard errors should be returned alongside the parameter estimates. Default is `FALSE`.
`response`	Integer vector indicating for which response variables the coefficients should be returned. If `NA`, coefficients for all response variables are returned.
`type`	Character vector specifying which type of coefficients should be returned. Possible values are `"beta"`, `"tau"`, `"power"` and `"correlation"`.
`...`	Additional arguments. Currently ignored and included for compatibility with the generic `coef` function.

Value

A `data.frame` with one row per parameter, containing:

- `Estimates`: parameter estimates;
- `Std.error`: standard errors (if requested);
- `Parameters`: parameter names;
- `Type`: parameter type;
- `Response`: response variable index.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

`confint.mcglm` *Confidence Intervals for Model Parameters*

Description

Computes Wald-type confidence intervals for parameter estimates from a fitted `mcglm` model, based on asymptotic normality.

Usage

```
## S3 method for class 'mcglm'
confint(object, parm, level = 0.95, ...)
```

Arguments

`object`	A fitted object of class `mcglm`.
`parm`	Optional specification of parameters for which confidence intervals are required. Can be a numeric vector of indices or a character vector of parameter names. If omitted, confidence intervals for all parameters are returned.
`level`	Numeric value giving the confidence level. Must be between 0 and 1. Default is `0.95`.
`...`	Additional arguments. Currently ignored and included for compatibility with the generic `confint` function.

Value

A numeric matrix with two columns corresponding to the lower and upper confidence limits. Rows correspond to model parameters.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

`ESS` *Generalized Error Sum of Squares*

Description

Extract the generalized error sum of squares (ESS) for objects of `mcglm` class.

Usage

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

Arguments

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

Value

An invisible list with a single numeric component:

ESS The generalized error sum of squares, scaled by the residual degrees of freedom.

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

`gof`, `plogLik`, `pAIC`, `pKLIC`, `GOSHO` and `RJC`.

`fitted.mcglm` *Fitted Values*

Description

Extract fitted (mean) values from a fitted `mcglm` model. For multivariate responses, fitted values are returned in matrix form, with one column per response variable.

Usage

```
## S3 method for class 'mcglm'
fitted(object, ...)
```

Arguments

`object`	A fitted object of class `mcglm`.
`...`	Additional arguments. Currently ignored and included for compatibility with the generic `fitted` function.

Value

A numeric matrix of fitted values. Rows correspond to observations and columns correspond to response variables.

Author(s)

Wagner Hugo Bonat, <`wbonat@ufpr.br`>

`fit_mcglm` *Chaser and Reciprocal Likelihood Algorithms*

Description

This function implements the two main algorithms used for fitting multivariate covariance generalized linear models. The chaser and the reciprocal likelihood algorithms.

Usage

```
fit_mcglm(list_initial, list_link, list_variance,
         list_covariance, list_X, list_Z, list_offset,
         list_Ntrial, list_power_fixed, list_sparse,
         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 `mc_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 `mc_variance_function` for details. Default `variance` = `"constant"`.
`list_covariance`	a list of covariance function names. Options are: `"identity"`, `"inverse"` and `"expm"`. Default `covariance` = `"identity"`.
`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`.
`list_sparse`	a list of logicals indicating if the matrices should be set up as sparse matrices. This argument is useful only when using exponential-matrix covariance link function. In the other cases the algorithm detects automatically if the matrix should be sparse or not.
`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 the results of the estimation procedure for multivariate covariance generalized linear models.

The object contains regression parameter estimates, covariance (dispersion) parameter estimates, fitted values, residuals and information about the iterative fitting process, such as the number of iterations, convergence status, sensitivity and variability matrices.

The returned object is intended for internal use. End users should rely on the output provided by `mcglm`, which wraps this function.

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

`mcglm`, `mc_matrix_linear_predictor`, `mc_link_function` and `mc_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 `mcglm` class.

Usage

```
gof(object)
```

Arguments

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

Value

A data frame with the following columns:

plogLik Numeric value of the pseudo Gaussian log-likelihood.

Df Integer giving the number of estimated parameters.

pAIC Numeric value of the pseudo Akaike Information Criterion.

pKLIC Numeric value of the pseudo Kullback–Leibler Information Criterion.

BIC Numeric value of the pseudo Bayesian 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.

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`.

`GOSHO` *Gosho Information Criterion*

Description

Extract the Gosho Information Criterion (GOSHO) for an object of `mcglm` class. WARNING: This function is limited to models with ONE response variable. This function is general useful only for longitudinal data analysis.

Usage

```
GOSHO(object, id, verbose = TRUE)
```

Arguments

`object`	an object of `mcglm` class.
`id`	a vector which identifies the clusters or groups. The length and order of id should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.
`verbose`	logical. Print or not the GOSHO value.

Value

An invisible list with a single numeric component:

Gosho The Gosho Information Criterion, computed for longitudinal data with a single response variable and scaled by the number of clusters.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

Source

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

See Also

`gof`, `plogLik`, `pAIC`, `pKLIC`, `ESS` and `RJC`.

`Hunting` *Hunting Data from Pico Basile, Bioko Island, Equatorial Guinea*

Description

This dataset contains a case study analyzed in Bonat et al. (2017) regarding animals hunted in the village of Basile Fang, Bioko Norte Province, Bioko Island, Equatorial Guinea. Monthly counts of blue duikers and other small animals shot or snared were collected from a random sample of 52 commercial hunters between August 2010 and September 2013. For each animal caught, the species, sex, capture method, and altitude were recorded. The dataset contains 1216 observations.

Usage

```
data(Hunting)
```

Format

A `data.frame` with 1216 observations and 11 variables:

`ALT` Factor with five levels indicating the altitude where the animal was caught.

`SEX` Factor with levels `Female` and `Male`.

`METHOD` Factor with levels `Escopeta` and `Trampa` indicating the method of capture.

`OT` Monthly number of other small animals hunted.

`BD` Monthly number of blue duikers hunted.

`OFFSET` Monthly number of hunter days.

`HUNTER` Hunter index.

`MONTH` Month index.

`MONTHCALENDAR` Month as calendar number (1 = January, ..., 12 = December).

`YEAR` Calendar year (2010–2013).

`HUNTER.MONTH` Index indicating observations taken for the same hunter and month.

Source

Bonat, W. H., et al. (2017). "Modelling the covariance structure in marginal multivariate count models: Hunting in Bioko Island." *Journal of Agricultural, Biological and Environmental Statistics*, 22(4):446–464.

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

Examples

```
library(mcglm)
library(Matrix)
data(Hunting, package = "mcglm")
formu <- OT ~ METHOD*ALT + SEX + ALT*poly(MONTH, 4)
Z0 <- mc_id(Hunting)
Z1 <- mc_mixed(~0 + HUNTER.MONTH, data = Hunting)
fit <- mcglm(linear_pred = c(formu),
             matrix_pred = list(c(Z0, Z1)),
             link = c("log"),
             variance = c("poisson_tweedie"),
             power_fixed = c(FALSE),
             control_algorithm = list(max_iter = 100),
             offset = list(log(Hunting$OFFSET)),
             data = Hunting)
summary(fit)
anova(fit)
```

`mcglm` *Fitting Multivariate Covariance Generalized Linear Models*

Description

Fits multivariate covariance generalized linear models. Models are specified through lists defining the linear predictors and matrix linear predictors. The user can choose among different link, variance, and covariance functions. Model fitting is based on an estimating function approach, combining quasi-score functions for regression parameters and Pearson estimating functions for covariance parameters.

Usage

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

Arguments

`linear_pred`	A list of model formulas, one for each response. See `formula` for details.
`matrix_pred`	A list of known matrices defining the matrix linear predictor for the covariance structure. See `mc_matrix_linear_predictor` for details.
`link`	A list of link function names, one for each response. Possible values are `"logit"`, `"probit"`, `"cauchit"`, `"cloglog"`, `"loglog"`, `"identity"`, `"log"`, `"sqrt"`, `"1/mu^2"`, and `"inverse"`.
`variance`	A list of variance function names. Possible values are `"constant"`, `"tweedie"`, `"poisson_tweedie"`, `"binomialP"`, and `"binomialPQ"`.
`covariance`	A list of covariance link function names. Possible values are `"identity"`, `"inverse"`, and `"expm"`.

`offset`	A list of numeric vectors specifying offsets for each response. Use `NULL` if no offset is required.
`Ntrial`	A list of numeric vectors specifying the number of trials for binomial responses. Only used for `binomialP` and `binomialPQ` variance functions.
`power_fixed`	A list of logical values indicating whether the power parameter should be fixed (`TRUE`) or estimated (`FALSE`).
`data`	a data frame.
`control_initial`	A list of initial values for the fitting algorithm. If set to `"automatic"`, initial values are generated internally using `mc_initial_values`.
`contrasts`	An optional list of contrasts passed to `model.matrix`.
`weights`	A list of numeric vectors of observation weights. Each element must have length equal to the number of observations. Missing observations should be coded as `NA`.
`control_algorithm`	A list of control parameters passed to the fitting algorithm. See `fit_mcglm` for details.

Value

An object of class `"mcglm"` representing a fitted multivariate covariance generalized linear model.

The returned object is a list produced by the fitting routine `fit_mcglm`, augmented with additional components used by post–estimation methods. The main components include:

beta_names A list of character vectors giving the names of the regression coefficients for each response variable.

power_fixed A list of logical values indicating whether the power parameters were fixed or estimated.

list_initial A list of initial values used in the fitting algorithm.

n_obs An integer giving the number of observations.

link A list of link functions used in the model.

variance A list of variance functions used in the model.

covariance A list of covariance link functions used in the model.

linear_pred A list of formulas defining the linear predictors.

matrix_pred A list of matrices defining the matrix linear predictors.

list_X A list of design matrices corresponding to the linear predictors.

observed A matrix of observed response values, with rows corresponding to observations and columns to response variables.

Ntrial A list containing the number of trials for each response variable, when applicable.

offset A list of offset vectors used in the model.

sparse A list of logical values indicating whether each matrix linear predictor is treated as sparse.

weights A numeric vector of weights used in the fitting process.

data The data frame used to fit the model.

con A list of control parameters used by the fitting algorithm.

Additional components may be present for internal use by methods such as `print`, `summary` and `predict`.

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_mcglm`, `mc_link_function` and `mc_variance_function`.

`mc_anova_disp` *Wald Tests for Dispersion Components*

Description

Performs Wald chi-square tests for dispersion (covariance) parameters by response variable in multivariate covariance generalized linear models fitted with `mcglm`. This function is intended for joint hypothesis testing of dispersion coefficients associated with categorical covariates with more than two levels. It is not designed for model comparison.

Usage

```
mc_anova_disp(object, idx_list, names_list, ...)
```

Arguments

`object`	An object of class `"mcglm"`, typically the result of a call to `mcglm`.
`idx_list`	A list of integer vectors indexing dispersion parameters to be jointly tested for each response.
`names_list`	A list of character vectors with covariate names to be displayed in the output tables.
`...`	Currently not used.

Value

The object is a list of data frames, one per response variable. Each data frame contains the following columns:

Covariate Name of the covariate associated with the dispersion parameters being tested.

Chi.Square Wald chi-square test statistic.

Df Degrees of freedom of the test.

p.value P-value associated with the chi-square test.

See Also

`mcglm`, `vcov`, `coef`

`mc_bias_corrected_std` *Bias-corrected Standard Error for Regression Parameters*

Description

Compute bias-corrected standard error for regression parameters in the context of clustered observations for an object of `mcglm` class. It is also robust and has improved finite sample properties.

Usage

```
mc_bias_corrected_std(object, id)
```

Arguments

`object`	an object of `mcglm` class.
`id`	a vector which identifies the clusters. The length and order of `id` should be the same as the number of observations. The data set are assumed to be sorted so that observations on a cluster are contiguous rows for all entities.

Value

A list with two elements. A vector of standard error and a variance-covariance matrix computed based on a bias corrected approach as described in the reference.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

Source

Nuamah, I. F. and Qu, Y. and Aminu, S. B. (1996). A SAS macro for stepwise correlated binary regression. Computer Methods and Programs in Biomedicine 49, 199–210.

See Also

`mc_robust_std`.

`mc_car` *Conditional Autoregressive Model Structure*

Description

The function `mc_car` helps to build the components of the matrix linear predictor used for fitting conditional autoregressive models. This function is used in general for fitting spatial areal data using the well known conditional autoregressive models (CAR). This function depends on a list of neighboors, such a list can be constructed, for example using the `tri2nb` function from the `spdep` package based on spatial coordinates. This way to specify the matrix linear predictor can also be applied for spatial continuous data, as an approximation.

Usage

```
mc_car(list_neigh, intrinsic = FALSE)
```

Arguments

`list_neigh`	list of neighboors.
`intrinsic`	logical.

Value

A list of a matrix (`intrinsic = TRUE`) or two matrices (`intrinsic = FALSE`). The main use of these matrices are as input in the `mcglm` function as linear covariance models in the argument `matrix_pred`.

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

`mc_id`, `mc_compute_rho`, `mc_conditional_test`, `mc_dist`, `mc_ma`, `mc_rw`
and `mc_mixed`.

`mc_complete_data` *Complete Data Set with NA*

Description

The function `mc_complete_data` completes a data set with NA values for helping to construct the components of the matrix linear predictor in models that require equal number of observations by subjects (id).

Usage

```
mc_complete_data(data, id, index, id.exp)
```

Arguments

`data`	a data.frame to be completed with NA.
`id`	name of the column (string) containing the subject id.
`index`	name of the column (string) containing the index to be completed.
`id.exp`	how the index is expected to be for all subjects.

Value

A data.frame with the same number of observations by subject. It is intended as a helper function to build the linear matrix predictor for models that require the same number of observations by subjects.

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

`mc_dglm`, `mc_ns`, `mc_ma` and `mc_rw`.

`mc_compute_rho` *Autocorrelation Estimates*

Description

Compute autocorrelation estimates based on a fitted model using the `mc_car` structure. The `mcglm` approach fits models using a linear covariance structure, but in general in this parametrization for spatial models the parameters have no simple interpretation in terms of spatial autocorrelation. The function `mc_compute_rho` computes the autocorrelation based on a fitted model.

Usage

```
mc_compute_rho(object, level = 0.975)
```

Arguments

`object`	an object or a list of objects representing a model of `mcglm` class.
`level`	the confidence level required.

Value

Returns a data frame with parameter estimate, standard error and confidential interval for the spatial autocorrelation parameter. It is used in the case of spatial models using the `mc_car` specification.

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

`mc_car` and `mc_conditional_test`.

`mc_conditional_test` *Conditional Hypotheses Tests*

Description

Compute conditional hypotheses tests for fitted `mcglm` model class. When fitting models with extra power parameters, the standard errors associated with the dispersion parameters can be large. In that cases, we suggest to conduct conditional hypotheses test instead of the orthodox marginal test for the dispersion parameters. The function `mc_conditional_test` offers an ease way to conduct such conditional test. Furthermore, the function is quite flexible and can be used for any other conditional hypotheses test.

Usage

```
mc_conditional_test(object, parameters, test, fixed)
```

Arguments

`object`	an object representing a model of `mcglm` class.
`parameters`	which parameters will be included in the conditional test.
`test`	index indicating which parameters will be tested given the values of the other parameters.
`fixed`	index indicating which parameters should be fixed on the conditional test.

Value

Returns a data frame with parameter estimates, conditional standard errors and Z-statistics. It is used to compute conditional hypothesis tests in mcglm fitted models.

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.

`mc_dglm` *Double Generalized Linear Models Structure*

Description

The function `mc_dglm` builds the components of the matrix linear predictor used for fitting double generalized linear models.

Usage

```
mc_dglm(formula, id, data)
```

Arguments

`formula`	a formula spefying the components of the covariance structure.
`id`	name of the column (string) containing the subject index. (If ts is not repeated measures, use id = 1 for all observations).
`data`	data set.

Value

A list containing diagonal matrices with entries defined by the covariates assumed to describe the matrix linear predictor. Each matrix corresponds to one component of the covariance model and is intended to be supplied to the `matrix_pred` argument of `mcglm`.

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

`mc_id`, `mc_dist`, `mc_ma`, `mc_rw`
and `mc_mixed`.

`mc_dist` *Distance Models Structure*

Description

The function `mc_dist` helps to build the components of the matrix linear predictor using matrices based on distances. This function is generaly used for the analysis of longitudinal and spatial data. The idea is to use the inverse of some measure of distance as for example the Euclidean distance to model the covariance structure within response variables. The model can also use the inverse of distance squared or high order power.

Usage

```
mc_dist(id, time, data, method = "euclidean")
```

Arguments

`id`	name of the column (string) containing the subject index. For spatial data use the same `id` for all observations (one unit sample).
`time`	name of the column (string) containing the index indicating the time. For spatial data use the same index for all observations.
`data`	data set.
`method`	distance measure to be used.

Details

The distance measure must be one of `"euclidean"`, `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` or `"minkowski"`. This function is a customize call of the `dist` function.

Value

A list containing a sparse matrix of class `dgCMatrix`. This matrix represents the design matrix for the linear predictor and is intended to be supplied to the `matrix_pred` argument of `mcglm`.

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

`dist`, `mc_id`, `mc_conditional_test`, `mc_car`, `mc_ma`, `mc_rw` and `mc_mixed`.

Examples

```
id <- rep(1:2, each = 4)
time <- rep(1:4, 2)
data <- data.frame("id" = id, "time" = time)
mc_dist(id = "id", time = "time", data = data)
mc_dist(id = "id", time = "time", data = data, method = "canberra")
```

`mc_id` *Independent Model Structure*

Description

Builds an identity matrix to be used as a component of the matrix linear predictor. It is in general the first component of the matrix linear predictor, a kind of intercept matrix.

Usage

```
mc_id(data)
```

Arguments

`data`	the data set to be used.

Value

A list with a single component:

Z0 A sparse identity matrix of class `ddiMatrix` with dimension equal to the number of observations, representing the intercept component of the matrix linear predictor.

It is intended to be supplied to the `matrix_pred` argument of `mcglm`.

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

`mc_dist`, `mc_ma`, `mc_rw` and `mc_mixed`.

`mc_initial_values` *Automatic Initial Values*

Description

This function provides initial values to be used when fitting multivariate covariance generalized linear models by using the function `mcglm`. 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 `mcglm` function. However, if the users want to change some of the initial values, this function can be useful.

Usage

```
mc_initial_values(linear_pred, matrix_pred, link, variance,
                  covariance, 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 `mc_matrix_linear_predictor` for details.
`link`	a list of link functions names, see `mcglm` for details.
`variance`	a list of variance functions names, see `mcglm` for details.
`covariance`	a list of covariance link functions names, see `mcglm` 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 `mc_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.

For the cases `covariance = "inverse"` and `covariance = "expm"` the inverse and the logarithm of the estimated dispersion parameter is used as initial value for the first component of the matrix linear predictor. 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

A list containing initial values for model parameters:

- **regression** A list of numeric vectors with initial values for the regression parameters of each response variable.
- **power** A list of numeric vectors with initial values for the power parameters associated with the variance functions.
- **tau** A list of numeric vectors with initial values for the dispersion and covariance-related parameters in the matrix linear predictor.
- **rho** A numeric vector with initial values for the correlation parameters between response variables.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

`mc_link_function` *Link Functions*

Description

The `mc_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

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

mc_logit(beta, X, offset)

mc_probit(beta, X, offset)

mc_cauchit(beta, X, offset)

mc_cloglog(beta, X, offset)

mc_loglog(beta, X, offset)

mc_identity(beta, X, offset)

mc_log(beta, X, offset)

mc_sqrt(beta, X, offset)
```

```
mc_invmu2(beta, X, offset)

mc_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 `mc_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 the following components:

mu A numeric vector of length n containing the mean response values obtained by applying the inverse link function to the linear predictor.

D A numeric matrix of dimension $n \times p$ containing the derivatives of μ with respect to the regression parameters β.

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)
mc_link_function(beta = c(1,0.5), X = X,
                 offset = NULL, link = 'log')
mc_link_function(beta = c(1,0.5), X = X,
                 offset = rep(10,5), link = 'identity')
```

`mc_ma` *Moving Average Model Structure*

Description

Builds components of the matrix linear predictor associated with moving average (MA) covariance structures. This function is mainly intended for longitudinal data analysis, but can also be used for time series data

Usage

```
mc_ma(id, time, data, order = 1)
```

Arguments

`id`	name of the column (string) containing the subject index. Note that this structure was designed to deal with longitudinal data. For times series data use the same `id` for all observations (one unit sample).
`time`	name of the column (string) containing the index indicating the time.
`data`	data set.
`order`	An integer specifying the order of the moving average process.

Details

This function was primarily designed for longitudinal data, but it can also be used for time series analysis. In this case, the `id` argument should contain a single identifier, representing one observational unit. Internally, the function constructs block-diagonal band matrices using `bandSparse`.

Value

A list with the following component:

Z1 A sparse matrix of class `nsCMatrix` representing the moving average component of the matrix linear predictor. The matrix has dimension equal to the total number of observations and is constructed as a block-diagonal matrix, with one block per subject (or time series), each block encoding a moving average structure of the specified order.

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

`mc_id`, `mc_dist`, `mc_car`, `mc_rw` and `mc_mixed`.

Examples

```
id <- rep(1:2, each = 4)
time <- rep(1:4, 2)
data <- data.frame("id" = id, "time" = time)
mc_ma(id = "id", time = "time", data = data, order = 1)
mc_ma(id = "id", time = "time", data = data, order = 2)
```

`mc_manova` *MANOVA-Type Test for Multivariate Covariance GLMs*

Description

Performs a MANOVA-type Wald test for multivariate covariance generalized linear models fitted using `mcglm`. The test is based on quadratic forms of the estimated regression parameters and their covariance matrix, yielding statistics analogous to the Hotelling–Lawley trace.

Usage

```
mc_manova(object, ...)
```

Arguments

`object`	An object of class `"mcglm"`.
`...`	Further arguments (currently not used).

Value

A data frame containing the MANOVA-type test results with the following columns:

Effects Names of the tested model effects.

Df Degrees of freedom associated with each effect.

Hotelling-Lawley Hotelling–Lawley trace statistic.

Chi-square Chi-square test statistic.

p-value P-values from the chi-square approximation.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

See Also

mcglm, coef.mcglm, vcov.mcglm

---

mc_manova_disp *MANOVA-Type Test for Dispersion Components of mcglm Models*

Description

Performs a MANOVA-type Wald test for the dispersion parameters of multivariate covariance generalized linear models fitted using mcglm. The test is based on quadratic forms of the estimated dispersion parameters and their covariance matrix, yielding statistics analogous to the Hotelling–Lawley trace.

Usage

```
mc_manova_disp(object, idx, effect_names, ...)
```

Arguments

`object`	An object of class "mcglm".
`idx`	An integer vector defining the grouping structure of dispersion parameters to be tested.
`effect_names`	A character vector with labels for the tested dispersion effects.
`...`	Further arguments (currently not used).

Value

A data frame containing the MANOVA-type test results for the dispersion parameters with the following columns:

Effects Names of the tested dispersion effects.

Df Degrees of freedom associated with each effect.

Hotelling-Lawley Hotelling–Lawley trace statistic.

Chi-square Chi-square test statistic.

p-value P-values from the chi-square approximation.

Author(s)

Wagner Hugo Bonat, <wbonat@ufpr.br>

See Also

mcglm, mc_manova, coef.mcglm, vcov.mcglm

`mc_matrix_linear_predictor`
Matrix Linear Predictor

Description

Computes the matrix linear predictor used in multivariate covariance generalized linear models. The matrix linear predictor is defined as a linear combination of known matrices weighted by dispersion parameters.

Usage

```
mc_matrix_linear_predictor(tau, Z)
```

Arguments

`tau`	A numeric vector of dispersion parameters.
`Z`	A list of known matrices with compatible dimensions.

Details

Given a list of known matrices $(Z_1, \dots, Z_D)$ and a vector of dispersion parameters $(\tau_1, \dots, \tau_D)$, this function computes their weighted sum. This object is typically used as a component of the matrix linear predictor in covariance modeling.

Value

A matrix of class `Matrix` representing the matrix linear predictor

$$U = \tau_1 Z_1 + \cdots + \tau_D Z_D.$$

The returned matrix has the same dimensions as the elements of `Z`. The returned object is intended for internal use only.

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 the Royal Statistical Society: Series C, 65:649–675.

See Also

`mc_id`, `mc_dist`, `mc_ma`, `mc_rw`, `mc_mixed`, `mc_car`

Examples

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

`mc_mixed` *Mixed Models Structure*

Description

Constructs the components of the matrix linear predictor associated with mixed-effects covariance structures in multivariate covariance generalized linear models. The function builds symmetric matrices representing variance and covariance components as functions of known covariates, following a linear mixed model formulation.

The `mc_mixed` function is primarily intended for repeated measures and longitudinal data, where observations are collected within a fixed number of groups, subjects, or experimental units.

Usage

```
mc_mixed(formula, data)
```

Arguments

`formula`	A model formula specifying the structure of the matrix linear predictor for the dispersion component. The first term must remove the intercept (`0 +`), and the second term must identify the grouping variable (e.g., subject or unit), which must be a `factor`. Additional covariates may be specified after a slash (`/`) to define random slopes and associated covariance components.
`data`	A `data.frame` containing all variables referenced in `formula`.

Details

The `formula` argument follows a syntax similar to that used for linear mixed models. The grouping variable must be provided as the second term in the formula and must be a `factor`; no internal coercion is performed. Covariates specified after the slash (`/`) may be continuous or categorical and define additional variance and covariance components. When only the grouping variable is specified (e.g., `~ 0 + SUBJECT`), the resulting structure corresponds to the compound symmetry covariance model.

By default, all pairwise interaction terms between components are included in the matrix linear predictor. Interaction terms may be excluded by removing the corresponding components from the resulting list.

Value

A list of symmetric sparse matrices of class `"dsCMatrix"`, each corresponding to a variance or covariance component of the matrix linear predictor for the dispersion structure. The list includes matrices associated with main effects and, by default, their pairwise interaction terms as implied by the mixed-effects specification in the formula. These matrices are used internally to construct the linear predictor of the covariance model in `mcglm`. It is intended to be supplied to the `matrix_pred` argument of `mcglm`.

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., et al. (2016). Modelling the covariance structure in marginal multivariate count models: Hunting in Bioko Island. Journal of Agricultural, Biological, and Environmental Statistics, 22(4), 446–464.

See Also

`mc_id`, `mc_conditional_test`, `mc_dist`, `mc_ma`, `mc_rw`, `mc_car`

Examples

```
SUBJECT <- gl(2, 6)
x1 <- rep(1:6, 2)
x2 <- rep(gl(2, 3), 2)
data <- data.frame(SUBJECT, x1, x2)

# Compound symmetry structure
mc_mixed(~ 0 + SUBJECT, data = data)

# Compound symmetry with random slope for x1
mc_mixed(~ 0 + SUBJECT/x1, data = data)

# Compound symmetry with random slopes for x1 and x2 and interactions
mc_mixed(~ 0 + SUBJECT/(x1 + x2), data = data)
```

`mc_ns` *Non-structured Covariance Model*

Description

Constructs the components of the matrix linear predictor associated with a fully non-structured covariance model in multivariate covariance generalized linear models. This specification allows each pair of observations within a unit to have its own covariance parameter, resulting in a highly flexible but parameter-intensive model.

Due to the quadratic growth in the number of parameters, this structure is typically suitable only for datasets with a small number of repeated measurements per unit.

Usage

```
mc_ns(id, data, group = NULL, marca = NULL)
```

Arguments

`id`	A character string giving the name of the column in `data` that identifies the observational units (e.g., subjects). Each unit must have the same number of observations. For time series or spatial data without replication, the same identifier should be used for all observations.
`data`	A `data.frame` containing the variables referenced by `id` and, optionally, `group`.
`group`	An optional character string giving the name of a column in `data` that defines groups for which different covariance structures may be specified. If `NULL`, a single non-structured covariance model is used for all units.
`marca`	An optional character string specifying the level of `group` for which the non-structured covariance components are excluded (i.e., set to zero). This allows selective activation of the non-structured covariance according to group membership.

Details

The function requires a balanced design, meaning that all units identified by `id` must have the same number of observations. An error is raised otherwise. When `group` and `marca` are provided, covariance components are generated only for units not belonging to the specified level `marca`; for those units, the corresponding blocks are set to zero.

Value

A list of symmetric block-diagonal matrices, each representing one covariance component of the non-structured matrix linear predictor. The length of the list is equal to $n(n-1)/2$, where n is the number of observations per unit. Each element of the list is a sparse matrix of class `"dgCMatrix"` obtained by stacking unit- specific covariance blocks along the diagonal. These matrices are used internally to construct the dispersion linear predictor in `mcglm`.

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

`mc_id`, `mc_dglm`, `mc_dist`, `mc_ma`, `mc_rw`, `mc_mixed`

`mc_robust_std` *Robust Standard Errors for Regression Parameters*

Description

Computes cluster-robust (sandwich-type) standard errors for the regression parameters of an object of class `mcglm`, accounting for within-cluster correlation.

Usage

```
mc_robust_std(object, id)
```

Arguments

`object`	An object of class `mcglm` representing a fitted marginal model.
`id`	An integer or factor vector identifying clusters or subjects. Its length and ordering must match the number and ordering of the observations used to fit the model.

Details

The robust variance–covariance matrix is obtained using an empirical estimator based on clustered residuals and the sensitivity matrix of the estimating equations. The implementation assumes that the data are correctly ordered such that observations belonging to the same cluster are stored in contiguous rows.

Value

A list with two components:

Std.Error A numeric vector containing the robust standard errors of the regression parameter estimates.

vcov A numeric matrix giving the robust variance–covariance matrix of the regression parameter estimates.

The returned objects are computed under the assumption that the data are in the correct cluster order.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

Source

Nuamah, I. F., Qu, Y., and Aminu, S. B. (1996). A SAS macro for stepwise correlated binary regression. *Computer Methods and Programs in Biomedicine*, 49, 199–210.

See Also

`mc_bias_correct_std`

`mc_rw` *Random Walk Model Structure*

Description

Constructs the components of the matrix linear predictor associated with random walk (RW) models for longitudinal or time series data. The user may specify the order of the random walk process.

Usage

```
mc_rw(id, time, data, order = 1, proper = FALSE)
```

Arguments

`id`	A character string giving the name of the column in `data` that identifies subjects or clusters.
`time`	A character string giving the name of the column in `data` that indexes time or ordering within each subject.
`data`	A data frame containing the variables specified in `id` and `time`.
`order`	A positive integer specifying the order of the random walk model.
`proper`	Logical indicating whether a proper random walk specification should be used.

Details

This function builds sparse precision matrix components corresponding to random walk structures of a given order. It is primarily intended for longitudinal data indexed by a subject identifier and a time variable. For pure time series data, the same `id` value should be used for all observations. When `proper = TRUE`, the precision structure is decomposed into diagonal and off-diagonal components.

Value

If `proper = FALSE`, a list with a single component:

Z1 A sparse matrix of class `dgCMatrix` representing the random walk precision structure.

If `proper = TRUE`, a list with two components:

Z1 A sparse diagonal matrix of class `dgCMatrix`.

Z2 A sparse off-diagonal matrix of class `dgCMatrix`.

The matrices are ordered consistently with the original data.

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

`mc_id`, `mc_dist`, `mc_car`, `mc_ma`, `mc_mixed`, `mc_compute_rho`

Examples

```
id <- rep(1:2, each = 4)
time <- rep(1:4, 2)
data <- data.frame(id = id, time = time)
mc_rw(id = "id", time = "time", data = data, order = 1, proper = FALSE)
mc_rw(id = "id", time = "time", data = data, order = 1, proper = TRUE)
mc_rw(id = "id", time = "time", data = data, order = 2, proper = TRUE)
```

`mc_sic` *Score Information Criterion for Regression Components*

Description

Computes the Score Information Criterion (SIC) for regression components of a fitted `mcglm` object. The SIC can be used for selecting covariates in the linear predictor and supports stepwise selection procedures.

Usage

```
mc_sic(object, scope, data, response, penalty = 2, weights)
```

Arguments

`object`	An object of class `mcglm`.
`scope`	A character vector with the names of covariates to be tested for inclusion in the linear predictor.
`data`	A data frame containing all variables involved in the model.
`response`	An integer indicating the response variable for which the SIC is computed.
`penalty`	A numeric penalty term applied to the SIC (default is 2).
`weights`	An optional numeric vector of weights used in model fitting. If not provided, unit weights are assumed.

Details

The SIC is computed using the quasi-score function associated with the regression parameters. For each candidate covariate in `scope`, the method evaluates its contribution via a score-based test statistic and applies a penalty for model complexity.

Value

A data frame with the following columns:

SIC Score Information Criterion value.

Covariates Name of the candidate covariate.

df Degrees of freedom associated with the test.

df_total Total number of regression parameters in the extended model.

Tu Score-based test statistic.

Chisq Reference chi-squared quantile with 95% confidence level.

References

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., et al. (2016). Modelling the covariance structure in marginal multivariate count models: Hunting in Bioko Island. *Journal of Agricultural, Biological and Environmental Statistics*, 22(4), 446–464.

See Also

`mc_sic_covariance`

Examples

```
set.seed(123)
x1 <- runif(100, -1, 1)
x2 <- gl(2, 50)
beta <- c(5, 0, 3)
X <- model.matrix(~ x1 + x2)
```

```
y <- rnorm(100, mean = X %*% beta, sd = 1)
data <- data.frame(y, x1, x2)

Z0 <- mc_id(data)
fit0 <- mcglm(
  linear_pred = c(y ~ 1),
  matrix_pred = list(Z0),
  data = data
)

mc_sic(fit0, scope = c("x1", "x2"), data = data, response = 1)
```

`mc_sic_covariance` *Score Information Criterion for Covariance Components*

Description

Computes the Score Information Criterion (SIC) for covariance components of a fitted `mcglm` object. The SIC-covariance is used to select components of the matrix linear predictor and can be employed in stepwise selection procedures.

Usage

```
mc_sic_covariance(object, scope, idx, data, penalty = 2, response, weights)
```

Arguments

`object`	An object of class `mcglm`.
`scope`	A list of matrices to be tested for inclusion in the matrix linear predictor.
`idx`	An integer vector indicating which matrices in `scope` belong to the same effect. This is useful when more than one matrix represents a single covariance component.
`data`	A data frame containing all variables involved in the model.
`penalty`	A numeric penalty term applied to the SIC (default is 2).
`response`	An integer indicating the response variable for which the SIC-covariance is computed.
`weights`	An optional numeric vector of weights used in model fitting. If not provided, unit weights are assumed.

Details

The SIC-covariance is computed using the Pearson estimating function. For each group of matrices defined by `idx`, a score-based test statistic is calculated to assess the contribution of the associated covariance components, penalized by model complexity.

Value

A data frame with the following columns:

SIC Score Information Criterion value.

df Degrees of freedom associated with the test.

df_total Total number of covariance parameters in the extended model.

Tu Score-based test statistic.

Chisq Reference chi-squared quantile with 95% confidence level.

References

Bonat, W. H., et al. (2016). Modelling the covariance structure in marginal multivariate count models: Hunting in Bioko Island. *Journal of Agricultural, Biological and Environmental Statistics*, 22(4), 446–464.

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

See Also

`mc_sic`

Examples

```
set.seed(123)
SUBJECT <- gl(10, 10)
y <- rnorm(100)
data <- data.frame(y, SUBJECT)

Z0 <- mc_id(data)
Z1 <- mc_mixed(~ 0 + SUBJECT, data = data)

fit0 <- mcglm(
  linear_pred = c(y ~ 1),
  matrix_pred = list(Z0),
  data = data
)

mc_sic_covariance(
  fit0,
  scope = Z1,
  idx = 1,
  data = data,
  response = 1
)
```

`mc_twin` *Twin Model Covariance Structures*

Description

Constructs the components of the matrix linear predictor for twin data analysis under ACDE-type models. The function generates covariance structures suitable for monozygotic (MZ) and dizygotic (DZ) twins and supports several biologically motivated and flexible model parameterizations.

Usage

```
mc_twin(id, twin.id, type, replicate = NULL, structure, data)

mc_twin_bio(id, twin.id, type, replicate = NULL, structure, data)

mc_twin_full(id, twin.id, type, replicate, formula, data)
```

Arguments

`id`	A string indicating the name of the column in `data` that identifies the twin pair. The same identifier must be shared by both twins in a pair.
`twin.id`	A string indicating the name of the column in `data` that identifies the twin within each pair. Typically coded as `1` and `2`.
`type`	A string indicating the name of the column in `data` that identifies the zygosity type. This variable must be a factor with exactly two levels: `"mz"` and `"dz"`, where `"mz"` is taken as the reference level.
`replicate`	An optional string indicating the name of the column in `data` that identifies replicated observations within the same twin pair, such as time points in longitudinal twin studies. If provided, it is treated as a factor.
`structure`	A string specifying the covariance structure to be constructed. Available options are `"full"`, `"flex"`, `"uns"`, `"ACE"`, `"ADE"`, `"AE"`, `"CE"` and `"E"`.
`data`	A data frame containing all variables referenced by the model.
`formula`	Internal argument used to define flexible and unstructured covariance models. Not intended for direct user specification.

Details

For biologically motivated structures (`"ACE"`, `"ADE"`, `"AE"`, `"CE"`, `"E"`), the function builds covariance matrices based on classical twin modeling assumptions. For flexible and unstructured options (`"full"`, `"flex"`, `"uns"`), the covariance structure is constructed using matrix linear predictors.

Value

A list of sparse matrices of class `dgCMatrix`, representing the components of the matrix linear predictor to be used in the `matrix_pred` argument of `mcglm`.

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

`mc_id`, `mc_dist`, `mc_car`, `mc_rw`, `mc_ns`, `mc_dglm`, `mc_mixed`.

`mc_variance_function` *Variance Functions for Generalized Linear Models*

Description

Computes the variance function and its derivatives with respect to regression, dispersion, and power parameters. This function supports standard power variance functions as well as binomial responses. Intended primarily for internal use in model fitting.

Usage

```
mc_variance_function(
  mu,
  power,
  Ntrial,
  variance,
  inverse,
  derivative_power,
  derivative_mu
)

mc_power(mu, power, inverse, derivative_power, derivative_mu)

mc_binomialP(mu, power, inverse, Ntrial,
                    derivative_power, derivative_mu)

mc_binomialPQ(mu, power, inverse, Ntrial,
                     derivative_power, derivative_mu)
```

Arguments

`mu`	Numeric vector of expected values. Typically obtained from `mc_link_function`.
`power`	Numeric value (for `power` and `binomialP`) or numeric vector of length two (for `binomialPQ`) representing the power parameters of the variance function.

`Ntrial`	Positive integer or numeric. Number of trials for binomial response variables.
`variance`	Character string specifying the variance function type: `"power"`, `"binomialP"`, or `"binomialPQ"`.
`inverse`	Logical. If `TRUE`, computes the inverse square root of the variance function.
`derivative_power`	Logical. If `TRUE`, computes the derivative with respect to the power parameter.
`derivative_mu`	Logical. If `TRUE`, computes the derivative with respect to `mu`.

Details

The function computes the variance function and its derivatives used in the estimation of generalized linear models for multiple response variables. For binomial responses, it accounts for the number of trials and supports both single (`binomialP`) and double (`binomialPQ`) power specifications.

Value

A named list containing one or more of the following elements, depending on the combination of logical arguments:

V_sqrt Square root of the variance function.

V_inv_sqrt Inverse square root of the variance function.

D_V_sqrt_power Derivative of V_sqrt with respect to the power parameter.

D_V_inv_sqrt_power Derivative of V_inv_sqrt with respect to the power parameter.

D_V_sqrt_mu Derivative of V_sqrt with respect to `mu`.

D_V_inv_sqrt_mu Derivative of V_inv_sqrt with respect to `mu`.

Author(s)

Wagner Hugo Bonat

Source

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

See Also

`mc_link_function`

Examples

```
x1 <- seq(-1, 1, length.out = 5)
X <- model.matrix(~x1)
mu <- mc_link_function(beta = c(1, 0.5), X = X, offset = NULL, link = "logit")
mc_variance_function(mu = mu$mu, power = c(2, 1), Ntrial = 1,
                     variance = "binomialPQ", inverse = FALSE,
                     derivative_power = TRUE, derivative_mu = TRUE)
```

`NewBorn` *Respiratory Physiotherapy on Premature Newborns*

Description

The NewBorn dataset is from a prospective study assessing the effect of respiratory physiotherapy on cardiopulmonary function in ventilated preterm newborn infants with birth weight less than 1500 g. The dataset was collected by the nursing team of Waldemar Monastier Hospital, Campo Largo, PR, Brazil, and analyzed in Bonat and Jorgensen (2016) as an example of mixed outcomes regression models.

Usage

```
data(NewBorn)
```

Format

A `data.frame` with 270 observations and 21 variables:

`Sex` Factor with levels `Female` and `Male`.

`GA` Gestational age in weeks.

`BW` Birth weight in grams.

`APGAR1M` APGAR index at the first minute of life.

`APGAR5M` APGAR index at the fifth minute of life.

`PRE` Factor indicating prematurity (YES/NO).

`HD` Factor indicating Hansen's disease (YES/NO).

`SUR` Factor indicating surfactant administration (YES/NO).

`JAU` Factor indicating jaundice (YES/NO).

`PNE` Factor indicating pneumonia (YES/NO).

`PDA` Factor indicating persistence of ductus arteriosus (YES/NO).

`PPI` Factor indicating primary pulmonary infection (YES/NO).

`OTHERS` Factor indicating other diseases (YES/NO).

`DAYS` Age in days.

`AUX` Factor indicating type of respiratory auxiliary (HOOD/OTHERS).

`RR` Respiratory rate (continuous).

`HR` Heart rate (continuous).

`SPO2` Oxygen saturation (bounded).

`TREAT` Factor with three levels: Respiratory physiotherapy, Evaluation 1, Evaluation 2, Evaluation 3.

`NBI` Newborn index.

`TIME` Days of treatment.

Source

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

Examples

```
library(mcglm)
library(Matrix)
data(NewBorn, package = "mcglm")

# Linear predictor example
formu <- SPO2 ~ Sex + APGAR1M + APGAR5M + PRE + HD + SUR
Z0 <- mc_id(NewBorn)
fit <- mcglm(
  linear_pred = c(formu),
  matrix_pred = list(Z0),
  link = "logit",
  variance = "binomialP",
  power_fixed = TRUE,
  data = NewBorn,
  control_algorithm = list(verbose = FALSE, tuning = 0.5)
)
summary(fit)
```

`pAIC` *Pseudo Akaike Information Criterion*

Description

Computes the pseudo Akaike information criterion (pAIC) for fitted multivariate covariance generalized linear models. The pAIC is defined as

$$pAIC = -2\,\ell_p + 2\,\mathrm{df},$$

where ℓ_p is the pseudo log-likelihood and df denotes the effective number of parameters in the model.

This criterion is intended for model comparison within the class of `mcglm` models fitted to the same data.

Usage

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

Arguments

`object`	An object of class `mcglm` or a list of such objects. When a list is provided, the pseudo log-likelihood is computed for each model.
`verbose`	Logical indicating whether the pAIC value should be printed to the console. Defaults to `TRUE`.

Details

The pAIC is based on the pseudo log-likelihood returned by `plogLik` and should be used with caution, as it does not correspond to a true likelihood-based information criterion. Comparisons are meaningful only for models fitted to the same response data.

Value

An (invisible) named list with a single element:

pAIC A numeric value giving the pseudo Akaike information criterion associated with the fitted model(s).

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`, `ESS`, `pKLIC`, `GOSHO`, `RJC`

`pBIC` *Pseudo Bayesian Information Criterion*

Description

Computes the pseudo Bayesian information criterion (pBIC) for fitted multivariate covariance generalized linear models. The pBIC is defined as

$$pBIC = -2\,\ell_p + \mathrm{df}\log(n),$$

where ℓ_p is the pseudo log-likelihood, df is the effective number of parameters, and n is the total number of observed responses.

This criterion provides a more strongly penalized alternative to `pAIC`, favoring more parsimonious models when comparing `mcglm` fits to the same data.

Usage

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

Arguments

`object`	An object of class `mcglm` or a list of such objects. When a list is supplied, the pseudo log-likelihood and the number of observations are computed by aggregating information across all models.
`verbose`	Logical indicating whether the pBIC value should be printed to the console. Defaults to `TRUE`.

Details

The sample size n used in the penalty term corresponds to the total number of observed responses, obtained from the `observed` component of the fitted `mcglm` object(s). As the pBIC is based on a pseudo log-likelihood, it should be used cautiously and only for relative comparisons among models fitted to the same data set.

Value

An (invisible) named list with a single element:

pBIC A numeric value giving the pseudo Bayesian information criterion associated with the fitted model(s).

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`, `ESS`, `pAIC`, `pKLIC`, `GOSHO`, `RJC`

`pKLIC` *Pseudo Kullback-Leibler Information Criterion*

Description

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

Usage

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

Arguments

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

Value

An invisible list with a single numeric component:

pKLIC The pseudo Kullback–Leibler Information Criterion computed from the pseudo log-likelihood and a penalty term based on the sensitivity and variability matrices.

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`, `ESS`, `pAIC`, `GOSHO` and `RJC`.

`plogLik` *Gaussian Pseudo-Loglikelihood*

Description

Computes the Gaussian pseudo-loglikelihood for fitted multivariate covariance generalized linear models. The pseudo-loglikelihood is obtained by assuming a multivariate normal distribution for the stacked response vector, using the estimated mean vector and covariance matrix from the fitted `mcglm` object.

Usage

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

Arguments

`object`	An object of class `mcglm` or a list of such objects. When a list is supplied, the pseudo-loglikelihood is computed for the joint model obtained by stacking the responses, fitted values and covariance matrices of all elements in the list.
`verbose`	Logical indicating whether the pseudo-loglikelihood value should be printed to the console. Defaults to `TRUE`.

Details

The Gaussian pseudo-loglikelihood is computed as

$$\ell_p = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(y-\mu)^\top\Sigma^{-1}(y-\mu),$$

where y is the stacked vector of observed responses, μ is the stacked vector of fitted means, and Σ is the estimated covariance matrix. For a list of `mcglm` objects, block-diagonal covariance matrices are constructed internally.

This quantity is used mainly for model comparison purposes and as a building block for pseudo-information criteria such as `pAIC` and `pBIC`. It is not a true likelihood unless the Gaussian assumption holds.

Value

An invisible list with the following components:

plogLik A numeric value giving the Gaussian pseudo-loglikelihood.

df An integer giving the total number of estimated parameters (degrees of freedom) used in the model.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

See Also

`pAIC`, `pBIC`, `ESS`, `pKLIC`, `GOSHO`, `RJC`

`plot.mcglm` *Diagnostic Plots for mcglm Objects*

Description

Produces diagnostic plots for fitted `mcglm` objects and returns the data used to generate the plots. Available diagnostics include:

- `"residuals"`: plots of fitted values vs. Pearson residuals and Q-Q plots of residuals;
- `"algorithm"`: plots of regression and covariance parameters and quasi-score iterations to inspect algorithm convergence;
- `"partial_residuals"`: partial residual plots for each covariate in the model.

Usage

```
## S3 method for class 'mcglm'
plot(
  x,
  type = c("residuals", "algorithm", "partial_residuals"),
  plot_graphics = TRUE,
  ...
)
```

Arguments

`x`	A fitted object of class `mcglm`.
`type`	Character string specifying the type of diagnostic plot to produce. One of `"residuals"`, `"algorithm"`, or `"partial_residuals"`.
`plot_graphics`	Logical; if `TRUE` (default), the function produces the plots. If `FALSE`, the function only returns the data used for plotting, which can be used to produce custom plots.
`...`	Additional arguments; currently ignored. Included for compatibility with the generic `plot` function.

Value

A list containing the data used to generate the plots:

- For `"residuals"`: `fitted` values and `residuals`.
- For `"algorithm"`: `IterationRegression`, `IterationCovariance`, `ScoreRegression`, `ScoreCovariance`.
- For `"partial_residuals"`: `residuals`, `list_beta`, and `partial_residuals` containing x and y values for each covariate.

Author(s)

Wagner Hugo Bonat, <`wbonat@ufpr.br`>

See Also

`residuals`, `fitted`, `plot`

Examples

```
library(mcglm)
set.seed(123)
mydata <- data.frame(y = rnorm(10), x1 = rnorm(10),
                     x2 = rbinom(10, size = 1, prob = 0.5))
Z0 <- mc_id(mydata)
fit <- mcglm(c(y ~ x1 + x2), matrix_pred = list(Z0), data = mydata)
# Produce plots and get data
diag_data <- plot(fit, type = "residuals")
# Only get data without plotting
diag_data <- plot(fit, type = "partial_residuals", plot_graphics = FALSE)
```

`print.mcglm` *Print Method for mcglm Objects*

Description

Prints a concise summary of a fitted `mcglm` object, including the model call, link and variance functions, regression coefficients and dispersion parameters for each response variable.

Usage

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

Arguments

`x`	A fitted object of class `mcglm`, typically returned by `mcglm()`.
`...`	Further arguments passed to or from other methods.

Value

No return value, called for its side effects.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

See Also

`print`, `summary.mcglm`

`residuals.mcglm` *Residuals for mcglm Objects*

Description

Computes residuals for a fitted `mcglm` object. Different types of residuals can be extracted, depending on the specified argument `type`.

Usage

```
## S3 method for class 'mcglm'
residuals(object, type = c("raw", "pearson", "standardized"), ...)
```

Arguments

`object`	An object of class `mcglm`.
`type`	A character string specifying the type of residuals to be returned. Options are: `"raw"` Raw residuals, defined as observed minus fitted values. `"pearson"` Pearson residuals, scaled by the marginal standard deviation. `"standardized"` Standardized residuals, obtained using the inverse covariance matrix.
`...`	Further arguments passed to or from other methods. Currently ignored.

Value

A numeric matrix of class `Matrix` with dimensions $n \times r$, where n is the number of observations and r is the number of response variables.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

See Also

`residuals`, `fitted.mcglm`

`RJC` *Rotnitzky–Jewell Information Criterion*

Description

Computes the Rotnitzky–Jewell information criterion (RJC) for objects of class `mcglm`. This criterion is based on quasi-likelihood theory and is intended for model assessment in marginal models.

Usage

```
RJC(object, id, verbose = TRUE)
```

Arguments

`object`	An object of class `mcglm` representing a fitted marginal model.
`id`	An integer or factor vector identifying the clusters. Its length and ordering must match the number and ordering of the observations used to fit the model.
`verbose`	Logical. If `TRUE`, the value of the RJC is printed to the console.

Details

The RJC is defined using the sensitivity and variability structures of the estimating equations and measures the discrepancy between them. The implementation assumes that the data are correctly ordered such that observations belonging to the same cluster are stored in contiguous rows.

Warning: This function is restricted to models with a single response variable.

Value

An invisible list with a single component:

RJC A numeric scalar giving the value of the Rotnitzky–Jewell information criterion.

Author(s)

Wagner Hugo Bonat, `<wbonat@ufpr.br>`

Source

Wang, M. (2014). Generalized estimating equations in longitudinal data analysis: A review and recent developments. *Advances in Statistics*, 1(1), 1–13.

See Also

`gof`, `plogLik`, `pAIC`, `pKLIC`, `ESS`, `GOSHO`

`soil` *Soil Chemistry Properties Dataset*

Description

Soil chemistry properties measured on a regular grid of 10 × 25 points, spaced by 5 meters. Each record contains the chemical composition and coordinates of the soil sample.

Usage

```
data(soil)
```

Format

A `data.frame` with 250 observations and 9 variables:

`COORD.X`	X coordinate of the sampling point.
`COORD.Y`	Y coordinate of the sampling point.
`SAND`	Proportion of sand in the soil sample.
`SILT`	Proportion of silt in the soil sample.
`CLAY`	Proportion of clay in the soil sample.
`PHWATER`	Soil pH measured in water.
`CA`	Calcium content of the soil.
`MG`	Magnesium content of the soil.
`K`	Potassium content of the soil.

Source

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

Examples

```
library(mcglm)
data(soil, package = "mcglm")

# Spatial model (tri2nb could be used but takes long)
Z1 <- mc_id(soil)

# Linear predictor example
form.ca <- CA ~ COORD.X*COORD.Y + SAND + SILT + CLAY + PHWATER
fit.ca <- mcglm(
  linear_pred = c(form.ca),
  matrix_pred = list(Z1),
  link = "log",
  variance = "tweedie",
  covariance = "inverse",
  power_fixed = TRUE,
  data = soil,
  control_algorithm = list(
    max_iter = 1000,
    tuning = 0.1,
    verbose = FALSE,
    tol = 1e-03
  )
)
```

`soya` *Soybeans Experiment Data*

Description

Dataset from an experiment conducted in a vegetation house with soybeans. Each plot contained two plants and the experiment involved three levels of soil water (`water`) and five levels of potassium fertilization (`pot`), arranged in five blocks (`block`). Three response variables are recorded: grain yield, number of seeds, and number of viable peas per plant. The dataset contains 75 observations and 7 variables.

Usage

```
data(soya)
```

Format

A `data.frame` with 75 observations and 7 variables:

`pot`	Factor with five levels of potassium fertilization.
`water`	Factor with three levels of amount of water in the soil.
`block`	Factor with five levels representing experimental blocks.
`grain`	Continuous variable representing grain yield per plant.
`seeds`	Count variable representing number of seeds per plant.
`viablepeas`	Binomial variable representing number of viable peas per plant.
`totalpeas`	Binomial variable representing total number of peas per plant.

Source

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

Examples

```
library(mcglm)
library(Matrix)
data(soya, package = "mcglm")

# Linear predictor example
formu <- grain ~ block + factor(water) * factor(pot)
Z0 <- mc_id(soya)
fit <- mcglm(linear_pred = c(formu), matrix_pred = list(Z0),
            data = soya)
anova(fit)
```

`summary.mcglm` *Summary for mcglm Objects*

Description

Produces a comprehensive summary of a fitted `mcglm` object. The summary includes estimates, standard errors, Wald Z statistics, and p-values for regression, dispersion, power, and correlation parameters. The function can either print the summary to the console or return it invisibly.

Usage

```
## S3 method for class 'mcglm'
summary(
  object,
  verbose = TRUE,
  print = c("Regression", "power", "Dispersion", "Correlation"),
  ...
)
```

Arguments

`object`	A fitted object of class `mcglm`.
`verbose`	Logical; if `TRUE` (default), prints the summary to the console. If `FALSE`, the summary is returned invisibly.
`print`	Character vector specifying which components of the summary to print. Possible values are `"Regression"`, `"Power"`, `"Dispersion"`, and `"Correlation"`. Default prints all components.
`...`	Further arguments passed to or from other methods. Currently ignored.

Details

The summary also prints information about the model fitting, including the link, variance, and covariance functions used, the algorithm method, any correction applied, and the number of iterations completed.

Value

Invisibly returns a list containing summary tables:

- For each response: `Regression`, `Power`, `Dispersion`.
- If applicable: `Correlation` summary.

Each table contains parameter estimates, standard errors, Wald Z values, and two-sided p-values.

Author(s)

Wagner Hugo Bonat, <`wbonat@ufpr.br`>

See Also

`print.mcglm`, `coef.mcglm`, `residuals.mcglm`

Examples

```
library(mcglm)
set.seed(123)
mydata <- data.frame(y = rnorm(10), x1 = rnorm(10),
                     x2 = rbinom(10, size = 1, prob = 0.5))
Z0 <- mc_id(mydata)
fit <- mcglm(c(y ~ x1 + x2), matrix_pred = list(Z0), data = mydata)
# Print full summary
summary(fit)
# Get summary invisibly
out <- summary(fit, verbose = FALSE)
```

`vcov.mcglm` *Variance-Covariance Matrix for mcglm Objects*

Description

Extracts the variance-covariance matrix of the estimated parameters from a fitted `mcglm` object.

Usage

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

Arguments

`object`	An object of class `mcglm`.
`...`	Further arguments passed to or from other methods. Currently ignored.

Value

A numeric matrix representing the variance-covariance matrix of all estimated model parameters. Row and column names correspond to the parameter identifiers.

Author(s)

Wagner Hugo Bonat, <`wbonat@ufpr.br`>

See Also

`coef.mcglm`, `summary.mcglm`

Index