Package: geer (via r-universe)

June 23, 2026

Type Package

Title Bias-Reduced and Penalized Generalized Estimating Equations

Version 0.1.0

Description Fits marginal regression models for repeated or clustered responses using generalized estimating equations (GEE). Provides ordinary GEE estimators, bias-reduced and bias-corrected GEE estimators, and Jeffreys-type penalized GEE estimators for binary, count and continuous responses. Methods are described in Touloumis (2026a) <doi:10.48550/arXiv.2606.16043> and Touloumis (2026b) <doi:10.48550/arXiv.2606.16058>.

License GPL-3

URL `https://github.com/AnestisTouloumis/geer`

BugReports `https://github.com/AnestisTouloumis/geer/issues`

Depends R (>= 4.0.0)

Imports brglm2, generics, Rcpp (>= 1.0.9)

LinkingTo Rcpp, RcppArmadillo

Suggests covr, emmeans (>= 2.0.0), estimability, testthat (>= 3.0.0), tibble

Encoding UTF-8

LazyData true

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Anestis Touloumis [aut, cre] (ORCID: <`https://orcid.org/0000-0002-5965-1639`>)

Maintainer Anestis Touloumis <`A.Touloumis@brighton.ac.uk`>

Repository https://cran.r-universe.dev

Date/Publication 2026-06-23 14:50:02 UTC

RemoteUrl https://github.com/cran/geer

RemoteRef HEAD

RemoteSha 40c7bc76a3b3653d87a77f2acb918cb20ffe7801

Contents

`geer-package` *Bias-Reduced and Penalized Generalized Estimating Equations*

Description

Fits marginal models for repeated or clustered responses using Generalized Estimating Equations (GEE). Supported estimation methods include the traditional GEE, bias-reducing GEE, bias-correcting GEE, and Jeffreys-prior penalized GEE. Continuous, binary and count responses are handled by `geewa`, while binary responses can also be handled by `geewa_binary` through an odds-ratio parameterization.

Author(s)

Anestis Touloumis <`A.Touloumis@brighton.ac.uk`>

References

Liang, K.Y. and Zeger, S.L. (1986) Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 13–22.

Touloumis, A. (2026) Jeffreys-Type Penalized GEE for Correlated Binary Data with an Odds-Ratio Parameterization. *Preprint*. `https://arxiv.org/abs/2606.16058`

Touloumis, A. (2026) Bias-Reduced GEE via Adjusted Estimating Equations, with Odds-Ratio Extensions. *Preprint*. `https://arxiv.org/abs/2606.16043`

See Also

Main functions:

- `geewa` for continuous, binary and count responses.
- `geewa_binary` for binary responses via an odds-ratio parameterization (preferred).
- `geer_control` for convergence and fitting options.
- `geecriteria` for model selection criteria.
- `summary.geer`, `tidy.geer`, and `glance.geer` for model summaries.
- `anova.geer`, `add1.geer`, `drop1.geer`, and `step_p` for model building.
- `vcov.geer`, `confint.geer`, `predict.geer`, `fitted.geer`, `residuals.geer`, and `frechet_bounds_cor` for inference and model diagnostics.

`add1.geer` *Add or Drop Single Terms to or from a geer Model*

Description

Computes all single terms in the `scope` argument that can be added to or dropped from a fitted `geer` model, fits the corresponding models, and returns a table summarizing the resulting changes in fit.

Usage

```
## S3 method for class 'geer'
add1(
  object,
  scope,
  test = c("wald", "score", "working-wald", "working-score", "working-lrt"),
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  pmethod = c("rao-scott", "satterthwaite"),
  ...
)

## S3 method for class 'geer'
drop1(
  object,
  scope,
  test = c("wald", "score", "working-wald", "working-score", "working-lrt"),
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  pmethod = c("rao-scott", "satterthwaite"),
  ...
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`scope`	a formula giving the terms to be considered for adding or dropping.
`test`	character string specifying the hypothesis testing procedure. Options are the Wald test (`"wald"`), the generalized score test (`"score"`), the modified working Wald test (`"working-wald"`), the modified working score test (`"working-score"`), and the modified working likelihood ratio test (`"working-lrt"`). Defaults to `"wald"`.
`cov_type`	character string specifying the covariance matrix estimator used for inference on the regression parameters. Options are the bias-corrected estimator (`"bias-corrected"`), the sandwich or robust estimator (`"robust"`), the degrees-of-freedom adjusted estimator (`"df-adjusted"`), and the model-based or naive estimator (`"naive"`). Defaults to `"bias-corrected"`.
`pmethod`	character string specifying the approximation used to compute the p-value for the modified working tests. Options are the Rao–Scott approximation (`"rao-scott"`) and the Satterthwaite approximation (`"satterthwaite"`). Defaults to `"rao-scott"`.
`...`	additional arguments passed to or from other methods.

Details

For `add1.geer()`, `scope` must specify the candidate terms to be added. If no eligible terms are supplied, an error is returned. In scope formulas, `.` denotes the set of terms already included in the model.

Model hierarchy is enforced: candidate additions and deletions must preserve marginality, so if a higher-order interaction is present, all of its lower-order component terms must also remain in the model.

Details of the hypothesis tests controlled by `test` are given in Rotnitzky and Jewell (1990). The option `test = "working-lrt"` is valid only when the model is fitted with an independence working association structure; otherwise an error is returned.

When `test %in% c("wald", "score")`, the `pmethod` argument is ignored and `cov_type` specifies the covariance estimator used to compute the test statistic. For modified working tests, `cov_type` determines the covariance matrix used to form the coefficients of the sum of independent chi-squared random variables, and `pmethod` specifies the approximation used to compute the p-value.

The output table also includes the Correlation Information Criterion (CIC) for each candidate model, which can be used to guide selection of the working association structure.

Value

An object of class `"anova"` summarizing the differences in fit between the models. For `add1.geer()` and `drop1.geer()`, the table contains one row for the current model and one row for each admissible single-term addition or deletion. Columns include `Df` (degrees of freedom of the test), `CIC` (Correlation Information Criterion), `Chi` (test statistic), and `Pr(>Chi)` (p-value).

References

Rotnitzky, A. and Jewell, N.P. (1990) Hypothesis testing of regression parameters in semiparametric generalized linear models for cluster correlated data. *Biometrika*, **77**, 485–497.

See Also

`anova.geer`, `step_p`, `geecriteria`, `geewa`, `geewa_binary`.

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", , drop = FALSE]

fitted_model <- geewa(
  formula = status ~ baseline + I(treatment == "active") + gender + visit + age,
  family = binomial(link = "probit"),
  data = respiratory2,
  id = id,
  repeated = visit,
  corstr = "ar1",
  method = "gee"
)
add1(
  fitted_model,
  scope = . ~ . + baseline:age + age:visit + I(treatment == "active"):age + age:gender,
  test = "score"
)

data("respiratory", package = "geer")
```

```
respiratory2 <- respiratory[respiratory$center == "C2", , drop = FALSE]

fitted_model <- geewa(
  formula = status ~ baseline + I(treatment == "active") + gender + visit + age,
  family = binomial(link = "probit"),
  data = respiratory2,
  id = id,
  repeated = visit,
  corstr = "ar1",
  method = "gee"
)
drop1(fitted_model, test = "score")
```

`anova.geer` *ANOVA Tables for geer Objects*

Description

Computes hypothesis test tables for one or more fitted `geer` objects, analogous to analysis-of-variance (ANOVA) tables for GLM models.

Usage

```
## S3 method for class 'geer'
anova(
  object,
  ...,
  test = c("wald", "score", "working-wald", "working-score", "working-lrt"),
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  pmethod = c("rao-scott", "satterthwaite")
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`...`	additional fitted model objects of class `"geer"`, used for sequential model comparisons.
`test`	character string specifying the hypothesis testing procedure. Options are the Wald test (`"wald"`), the generalized score test (`"score"`), the modified working Wald test (`"working-wald"`), the modified working score test (`"working-score"`), and the modified working likelihood ratio test (`"working-lrt"`). Defaults to `"wald"`.
`cov_type`	character string specifying the covariance matrix estimator used for inference on the regression parameters. Options are the bias-corrected estimator (`"bias-corrected"`), the sandwich or robust estimator (`"robust"`), the degrees-of-freedom adjusted estimator (`"df-adjusted"`), and the model-based or naive estimator (`"naive"`). Defaults to `"bias-corrected"`.

`pmethod`	character string specifying the approximation used to compute the p-value for the modified working tests. Options are the Rao–Scott approximation (`"rao-scott"`) and the Satterthwaite approximation (`"satterthwaite"`). Defaults to `"rao-scott"`.

Details

Details of the hypothesis tests controlled by `test` are given in Rotnitzky and Jewell (1990). The option `test = "working-lrt"` is valid only when the model is fitted with an independence working association structure; otherwise an error is returned.

When `test %in% c("wald", "score")`, the `pmethod` argument is ignored and `cov_type` specifies the covariance estimator used to compute the test statistic. For modified working tests, `cov_type` determines the covariance matrix used to form the coefficients of the sum of independent chi-squared random variables, and `pmethod` specifies the approximation used to compute the p-value.

When comparing two or more models, the data must be identical across all fits, and the models must be nested in the order supplied. In particular, each consecutive pair of models must be nested.

Value

An object of class `c("anova", "data.frame")`. With a single model, the table reports tests for terms added sequentially (first to last). With multiple models, the table reports sequential comparisons between each consecutive pair of nested models. Columns include `Df` (degrees of freedom of the test), `Resid. Df` (residual degrees of freedom), `Chi` (test statistic), and `Pr(>Chi)` (p-value). Unlike `add1.geer` and `drop1.geer`, no `CIC` column is included; use `geecriteria` for working association structure selection.

References

Rotnitzky, A. and Jewell, N.P. (1990) Hypothesis testing of regression parameters in semiparametric generalized linear models for cluster correlated data. *Biometrika*, **77**, 485–497.

See Also

`add1.geer`, `drop1.geer` for type II tests where each term is dropped one at a time while respecting model hierarchy; `step_p` for stepwise model selection; `geecriteria` for model comparison criteria.

Examples

```
data("cerebrovascular", package = "geer")

## Single-model ANOVA (sequential terms)
fit_full <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
anova(fit_full, test = "wald", cov_type = "robust")
```

```
## Two-model comparison (models must be nested)
fit_null <- geewa_binary(
  formula = ecg ~ 1,
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
anova(fit_null, fit_full, test = "wald", cov_type = "robust")
```

`cerebrovascular` *Cerebrovascular Deficiency Trial*

Description

Data from a randomized two-period crossover trial on cerebrovascular deficiency.

Usage

```
cerebrovascular
```

Format

A data frame with 134 rows and 4 columns:

id integer subject identifier.

period integer period identifier.

ecg binary indicator of ECG status, 0 = abnormal and 1 = normal.

treatment factor with levels `active` and `placebo`.

Details

Sixty-seven subjects were enrolled in a two-period crossover trial, yielding 134 observations (two per subject). Subjects were randomly assigned to receive either placebo followed by active treatment, or active treatment followed by placebo. ECG status was classified as normal or abnormal at each period.

References

Jones, B. and Kenward, M.G. (1989) *Design and Analysis of Cross-over Trials*. London: Chapman and Hall/CRC Press.

Examples

```
data("cerebrovascular", package = "geer")
str(cerebrovascular)
```

`cholecystectomy` *Shoulder Pain After Laparoscopic Cholecystectomy Trial*

Description

Data from a randomized clinical trial on shoulder pain after laparoscopic cholecystectomy.

Usage

```
cholecystectomy
```

Format

A data frame with 246 rows and 6 columns:

id integer subject identifier.

time integer time identifier.

pain binary indicator of shoulder pain status, 0 = high pain and 1 = low pain.

treatment factor with levels `active` and `placebo`.

gender factor with levels `female` and `male`.

age numeric age in years.

Details

Forty-one subjects were enrolled in a randomized clinical trial of shoulder pain after laparoscopic cholecystectomy, yielding 246 observations (six per subject). Subjects were assigned to either the active group (with abdominal suction) or the placebo group (without abdominal suction). Shoulder pain was assessed on six occasions: morning and afternoon on each of the first three postoperative days.

Source

Lumley, T. (1996) Generalized estimating equations for ordinal data: a note on working correlation structures. *Biometrics*, **52**, 354–361.

References

Jorgensen, J.O., Gillies, R.B., Hunt, D.R., Caplehorn, J.R.M. and Lumley, T. (1995) A simple and effective way to reduce postoperative pain after laparoscopic cholecystectomy. *Australian and New Zealand Journal of Surgery*, **65**, 466–469.

Examples

```
data("cholecystectomy", package = "geer")
str(cholecystectomy)
```

`coef.geer` *Extract Model Coefficients from a geer Object*

Description

Extracts the estimated regression coefficients from a fitted `geer` object. `coefficients` is an alias for `coef`.

Usage

```
## S3 method for class 'geer'
coef(object, ...)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`...`	additional arguments passed to or from other methods.

Value

A named numeric vector of estimated regression coefficients. The names correspond to the columns of the model matrix.

See Also

`vcov.geer`, `confint.geer`, `summary.geer`.

Examples

```
data("leprosy", package = "geer")
fit <- geewa(
  formula = bacilli ~ factor(period) + factor(period):treatment,
  family = poisson(link = "log"),
  data = leprosy,
  id = id
)
coef(fit)

data("cerebrovascular", package = "geer")
fit_bin <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id
)
coef(fit_bin)
```

`confint.geer` *Confidence Intervals for Model Parameters from a geer Object*

Description

Computes Wald-type confidence intervals for one or more regression parameters from a fitted `geer` object.

Usage

```
## S3 method for class 'geer'
confint(
  object,
  parm,
  level = 0.95,
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  ...
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
`level`	the confidence level required.
`cov_type`	character string specifying the covariance matrix estimator used for inference on the regression parameters. Options are the bias-corrected estimator (`"bias-corrected"`), the sandwich or robust estimator (`"robust"`), the degrees-of-freedom adjusted estimator (`"df-adjusted"`), and the model-based or naive estimator (`"naive"`). Defaults to `"bias-corrected"`.
`...`	additional arguments passed to or from other methods.

Details

Confidence intervals are computed as $\hat{\beta} \pm z_{1-\alpha/2}\,\mathrm{SE}(\hat{\beta})$, where standard errors are obtained from `vcov(object, cov_type = cov_type)`. The covariance estimator used is controlled by `cov_type`; see `vcov.geer` for details. The resulting intervals rely on the usual large-sample normal approximation.

Value

A matrix with columns giving lower and upper confidence limits for each parameter. The columns are labeled by the corresponding tail probabilities in percent, for example `"2.5%"` and `"97.5%"` when `level = 0.95`.

See Also

`vcov.geer`, `coef.geer`, `tidy.geer`.

Examples

```
data("cerebrovascular", package = "geer")
fit <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
confint(fit)
confint(fit, parm = "treatmentactive")
confint(fit, cov_type = "naive")
```

`depression` *Postnatal Depression Oestrogen Patch Trial*

Description

Data from a randomized clinical trial on the efficacy of oestrogen patches in treating postnatal depression.

Usage

```
depression
```

Format

A data frame with 366 rows and 5 columns:

id integer subject identifier.

visit integer follow-up month identifier.

score numeric Edinburgh Postnatal Depression Scale (EPDS) score. Higher scores indicate greater depression.

treatment factor with levels `placebo` and `oestrogen`.

baseline numeric EPDS score at baseline.

Details

Sixty-one women with major depression were randomly assigned to either a placebo control group or an oestrogen patch group, yielding 366 observations (six monthly assessments per subject). Before treatment, depressive symptoms were assessed using the Edinburgh Postnatal Depression Scale (EPDS). EPDS scores were then collected monthly for six months.

Source

https://stats.oarc.ucla.edu/spss/library/spss-librarypanel-data-analysis-using-gee/

References

Gregoire, A.J.P., Kumar, R., Everitt, B., Henderson, A.F. and Studd, J.W.W. (1996) Transdermal oestrogen for treatment of severe postnatal depression. *The Lancet*, **347**, 930–933.

Examples

```
data("depression", package = "geer")
str(depression)
```

`emmeans-support` *emmeans Support for geer Objects*

Description

Methods that allow fitted `geer` objects to work with the **emmeans** package.

These functions are used internally by **emmeans** and are not usually called directly by end users. Once both packages are installed, functions such as `emmeans::emmeans()` and `emmeans::ref_grid()` can be applied to fitted `geer` objects.

By default, **emmeans** calculations for `geer` objects use the bias-corrected covariance matrix. Alternative covariance estimators may be requested via `vcov.method`, with supported character values `"bias-corrected"`, `"robust"`, `"df-adjusted"`, and `"naive"`.

In line with the large-sample inference used elsewhere in **geer**, these methods use asymptotic inference with degrees of freedom equal to `Inf`.

Usage

```
recover_data.geer(object, data = NULL, ...)

emm_basis.geer(
  object,
  trms,
  xlev,
  grid,
  vcov.method = "bias-corrected",
  cov_type = NULL,
  vcov. = NULL,
  misc = NULL,
  ...
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`data`	optional data frame used by **emmeans** to recover the model predictors.
`...`	additional arguments passed through from **emmeans**.
`trms`	the terms component supplied by **emmeans**.
`xlev`	the factor levels supplied by **emmeans**.
`grid`	the reference grid supplied by **emmeans**.
`vcov.method`	covariance specification to use for **emmeans** calculations. This may be a character string specifying one of the supported covariance estimators `"bias-corrected"`, `"robust"`, `"df-adjusted"`, `"naive"`, or a covariance matrix or function accepted by **emmeans** through its `vcov.` mechanism. Defaults to `"bias-corrected"`.
`cov_type`	optional alias for `vcov.method`.
`vcov.`	optional covariance matrix or function supplied through the standard **emmeans** `vcov.` argument. When provided, it takes precedence over `vcov.method`.
`misc`	optional list passed through by **emmeans**.

Value

`recover_data.geer()` returns a recovered predictor data frame for use by **emmeans**. `emm_basis.geer()` returns the basis components required by **emmeans** to construct an `emmGrid` object.

See Also

`vcov.geer`, `geewa`, `geewa_binary`.

Examples

```
if (requireNamespace("emmeans", quietly = TRUE)) {
  data("cerebrovascular", package = "geer")

  fit <- geewa_binary(
    formula = ecg ~ treatment + factor(period),
    link = "logit",
    data = cerebrovascular,
    id = id,
    orstr = "exchangeable"
  )

  emmeans::emmeans(fit, ~ treatment)
  emmeans::emmeans(fit, ~ treatment, type = "response")
  emmeans::emmeans(fit, ~ treatment, vcov.method = "naive")
}
```

`epilepsy` *Progabide Epilepsy Trial*

Description

Data from a randomized clinical trial on the efficacy of progabide in treating partial seizures.

Usage

```
epilepsy
```

Format

A data frame with 236 rows and 6 columns:

id integer subject identifier.

visit integer visit identifier corresponding to a two-week interval.

seizures integer count of epileptic seizures.

treatment factor with levels `placebo` and `progabide`.

lnbaseline numeric logarithm of one-quarter of the number of seizures in the baseline 8-week interval.

lnage numeric logarithm of age in years.

Details

Fifty-nine subjects with partial seizures were enrolled in a randomized clinical trial, yielding 236 observations (four visits per subject). Subjects were assigned to receive either progabide or placebo. Seizure counts were recorded during the 8 weeks before treatment (baseline). After treatment, seizure counts were assessed at four 2-week clinic visits.

Source

Thall, P.F. and Vail, S.C. (1990) Some covariance models for longitudinal count data with overdispersion. *Biometrics*, **46**, 657–671.

References

Carey, V.J. and Wang, Y.G. (2011) Working covariance model selection for generalized estimating equations. *Statistics in Medicine*, **30**, 3117–3124.

Examples

```
data("epilepsy", package = "geer")
str(epilepsy)
```

`fitted.geer` *Extract Model Fitted Values from a geer Object*

Description

Extracts the fitted mean values on the response scale from a fitted `geer` object. `fitted.values` is an alias for `fitted`.

Usage

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

Arguments

`object`	a fitted model object of class `"geer"`.
`...`	additional arguments passed to or from other methods.

Value

A numeric vector of fitted mean values on the response scale, of the same length as the number of observations used in fitting.

See Also

`residuals.geer`, `predict.geer`.

Examples

```
data("leprosy", package = "geer")
fit <- geewa(
  formula = bacilli ~ factor(period) + factor(period):treatment,
  family = poisson(link = "log"),
  data = leprosy,
  id = id
)
head(fitted(fit))
```

`frechet_bounds_cor` *Frechet Bounds for a Working Correlation Matrix*

Description

For a fitted `geer` model from `geewa` with a `binomial` family and a non-independence association structure, checks whether each off-diagonal entry of the working correlation matrix lies within the Frechet bounds implied by the fitted marginal probabilities. Results are summarized at the time-pair level.

Usage

```
frechet_bounds_cor(object)
```

Arguments

`object`	an object of class `geer` fitted via `geewa` with `family = binomial()` and a non-independence association structure.

Details

For a pair of observations at times j and k within cluster i, with fitted marginal probabilities π_{ij} and π_{ik}, the Frechet bounds on their correlation are

$$\ell_{ijk} = \max\left(-\sqrt{\frac{\pi_{ij}\pi_{ik}}{(1-\pi_{ij})(1-\pi_{ik})}},\ -\sqrt{\frac{(1-\pi_{ij})(1-\pi_{ik})}{\pi_{ij}\pi_{ik}}}\right)$$

$$u_{ijk} = \min\left(\sqrt{\frac{\pi_{ij}(1-\pi_{ik})}{\pi_{ik}(1-\pi_{ij})}},\ \sqrt{\frac{\pi_{ik}(1-\pi_{ij})}{\pi_{ij}(1-\pi_{ik})}}\right)$$

The working correlation value `cor` for a time pair (j,k) is the same for all clusters and is read from the fitted working correlation matrix. The bounds ℓ_{ijk} and u_{ijk} vary across clusters because they depend on the cluster-specific fitted probabilities. The tightest bounds across clusters, lower_max (maximum lower bound) and upper_min (minimum upper bound), are reported in the returned data frame. The column `n_violated` counts the number of clusters for which `cor` falls outside $(\ell_{ijk},\, u_{ijk})$.

Value

a data frame with one row per unique time pair (j,k) and columns:

- `alpha_name` label of the form `alpha_j.k` identifying the working correlation entry for this time pair.
- `alpha_value` working correlation value for the time pair.
- `lower_max` maximum Frechet lower bound across clusters, giving the tightest lower admissibility constraint.

`upper_min` minimum Frechet upper bound across clusters, giving the tightest upper admissibility constraint.

`n_violated` number of clusters for which `alpha_value` falls outside the cluster-specific Frechet bounds.

See Also

`geewa`.

Examples

```
data("cholecystectomy", package = "geer")

fit <- geewa(
  formula = pain ~ treatment + gender + age + I(time >4),
  family = binomial(link = "logit"),
  data = cholecystectomy,
  id = id,
  repeated = time,
  corstr = "unstructured",
  method = "gee"
)
frechet_bounds_cor(fit)
```

`geecriteria` *Model Selection Criteria for geer Objects*

Description

Computes model selection criteria for one or more fitted `geer` objects, supporting both marginal mean model comparison and working association structure selection.

Usage

```
geecriteria(
  object,
  ...,
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  digits = 2
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`...`	additional fitted model objects of class `"geer"` to be included in the comparison.

`cov_type`	character string specifying the covariance estimator used in the covariance-based criteria. Options are the bias-corrected estimator (`"bias-corrected"`), the sandwich or robust estimator (`"robust"`), the degrees-of-freedom adjusted estimator (`"df-adjusted"`), and the model-based or naive estimator (`"naive"`). Defaults to `"bias-corrected"`.
`digits`	non-negative integer giving the number of decimal places used to round the reported criteria. Defaults to `2`.

Details

The reported criteria are:

`QIC` Quasi Information Criterion for comparing marginal mean models. Smaller values are preferred.

`CIC` Correlation Information Criterion, used here for selecting the working association structure. Smaller values are preferred.

`RJC` Rotnitzky and Jewell Criterion. Smaller values are preferred.

`QICu` A variant of QIC primarily intended for comparing marginal mean models with different covariate sets. Smaller values are preferred.

`GESSC` Generalized Error Sum of Squares Criterion. Smaller values are preferred.

`GPC` Gaussian Pseudolikelihood Criterion. Unlike the other criteria, larger values are preferred, because GPC is a pseudolikelihood measure rather than an information criterion.

`Parameters` Number of regression parameters in the marginal mean model.

The `cov_type` argument affects only the covariance-based criteria `QIC`, `CIC`, and `RJC`; the remaining criteria are unaffected.

If the supplied models do not all have the same number of observations, a warning is issued.

Value

A data frame with one row per fitted model and columns `QIC`, `CIC`, `RJC`, `QICu`, `GESSC`, `GPC`, and `Parameters`, as described in the Details section. When more than one model is supplied, row names are set to the deparsed model expressions.

References

Carey, V.J. and Wang, Y.G. (2011) Working covariance model selection for generalized estimating equations. *Statistics in Medicine*, **30**, 3117–3124.

Chaganty, N.R. and Shults, J. (1999) On eliminating the asymptotic bias in the quasi-least squares estimate of the correlation parameter. *Journal of Statistical Planning and Inference*, **76**, 145–161.

Hin, L.Y. and Wang, Y.G. (2009) Working correlation structure identification in generalized estimating equations. *Statistics in Medicine*, **28**, 642–658.

Pan, W. (2001) Akaike's information criterion in generalized estimating equations. *Biometrics*, **57**, 120–125.

Rotnitzky, A. and Jewell, N.P. (1990) Hypothesis testing of regression parameters in semiparametric generalized linear models for cluster correlated data. *Biometrika*, **77**, 485–497.

See Also

glance.geer, anova.geer, geewa, geewa_binary.

Examples

```
## Single model
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  corstr = "exchangeable"
)
geecriteria(fit)

## Compare working correlation structures
fit_ind <- update(fit, corstr = "independence")
fit_ar1 <- update(fit, corstr = "ar1")
geecriteria(fit_ind, fit, fit_ar1)

## Compare estimation methods
data("cerebrovascular", package = "geer")
fitted_gee <- geewa_binary(
  formula = ecg ~ factor(period) * treatment,
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable",
  method = "gee"
)
fitted_brgee <- update(fitted_gee, method = "brgee-robust")
geecriteria(fitted_gee, fitted_brgee, cov_type = "robust")
```

`geer_control` *Control Parameters for geer Model Fitting*

Description

Creates a list of control parameters for iterative fitting in geewa and geewa_binary. This function is used internally, but may also be supplied directly through the `control` argument of those functions.

Usage

```
geer_control(
  tolerance = 1e-06,
  maxiter = 500,
  or_adding = 0.5,
```

```
  step_maxiter = 10,
  step_multiplier = 1,
  jeffreys_power = 0.5
)
```

Arguments

`tolerance`	strictly positive convergence tolerance. Defaults to `1e-6`.
`maxiter`	positive integer giving the maximum number of fitting iterations. Defaults to `500`.
`or_adding`	strictly positive continuity-correction constant added to each cell of the marginalized 2×2 contingency tables used to estimate working odds ratios in `geewa_binary`. Defaults to `0.5`. Ignored by `geewa`.
`step_maxiter`	positive integer giving the maximum number of step-halving attempts allowed within an iteration. Defaults to `10`.
`step_multiplier`	positive integer used to scale the proposed step before step-halving begins. A value greater than `1` enlarges the initial step; the default of `1` leaves the scoring step unscaled.
`jeffreys_power`	strictly positive constant giving the power of the Jeffreys-prior penalty. Defaults to `0.5`, which corresponds to the GEE analogue of the standard Jeffreys prior.

Details

The `jeffreys_power` argument is used only when `method %in% c("pgee-jeffreys", "opgee-jeffreys", "hpgee-jeffreys")` in `geewa` or `geewa_binary`.

Value

A named list with elements `tolerance`, `maxiter`, `or_adding`, `step_maxiter`, `step_multiplier`, and `jeffreys_power`, suitable for use as the `control` argument to `geewa` or `geewa_binary`.

References

Touloumis, A. (2026) Jeffreys-prior penalized GEE for correlated binary data with an odds-ratio parameterization. Preprint.

See Also

`geewa`, `geewa_binary`.

Examples

```
## Default control parameters
geer_control()

## Tighter convergence tolerance and fewer maximum iterations
geer_control(tolerance = 1e-8, maxiter = 200)
```

```
## Custom continuity correction for odds-ratio estimation
geer_control(or_adding = 1)

## Weaker Jeffreys-prior penalty for Jeffreys-type fits
geer_control(jeffreys_power = 0.1)
```

`geewa` *Fitting (Adjusted) Generalized Estimating Equations*

Description

Fits a marginal model for repeated or clustered responses using Generalized Estimating Equations (GEE). Supported estimation methods include the traditional GEE, bias-reducing GEE, bias-correcting GEE, and Jeffreys-prior penalized GEE.

Usage

```
geewa(
  formula,
  family = gaussian(link = "identity"),
  data = parent.frame(),
  id,
  repeated,
  control = geer_control(...),
  corstr = "independence",
  Mv = 1,
  method = "gee",
  weights,
  beta_start = NULL,
  offset,
  control_glm = list(...),
  use_p = TRUE,
  alpha_vector = NULL,
  phi_fixed = FALSE,
  phi_value = 1,
  ...
)
```

Arguments

`formula`	`formula` expression of the form `response` ~ `predictors`: a symbolic description of the marginal model to be fitted.
`family`	a `family` object specifying the marginal variance and link functions. Supported families are `gaussian`, `binomial`, `poisson`, `Gamma`, `inverse.gaussian`, `quasi`, `quasibinomial` and `quasipoisson`. Defaults to `gaussian(link = "identity")`.
`data`	optional data frame containing variables referenced in `formula`, `id` and `repeated`.

`id`	variable identifying the clusters.
`repeated`	optional variable identifying the order of observations within each cluster.
`control`	a `geer_control` list specifying convergence tolerance, maximum iterations, step-halving parameters, and the Jeffreys-prior power. Defaults to `geer_control()`.
`corstr`	character string specifying the working correlation structure. Options are `"independence"`, `"exchangeable"`, `"ar1"`, `"m-dependent"`, `"unstructured"`, `"toeplitz"` and `"fixed"`. Defaults to `"independence"`.
`Mv`	positive integer giving the number of lags for the m-dependent working correlation structure. Defaults to `1` (lag-1 dependence). Must be set explicitly when lags other than 1 are intended. Ignored when `corstr != "m-dependent"`.
`method`	character string specifying the estimation method. Options are the traditional GEE (`"gee"`), bias-reducing methods (`"brgee-robust"`, `"brgee-empirical"`, `"brgee-naive"`), bias-corrected methods (`"bcgee-robust"`, `"bcgee-empirical"`, `"bcgee-naive"`), the fully iterated Jeffreys-prior penalized GEE (`"pgee-jeffreys"`), the one-step penalized GEE (`"opgee-jeffreys"`), and the hybrid one-step GEE (`"hpgee-jeffreys"`). Defaults to `"gee"`.
`weights`	optional numeric vector of observation weights. Must be finite and strictly positive. If not supplied, all weights are 1.
`beta_start`	optional numeric vector of starting values for the regression parameters. If `NULL` (default), starting values are computed by fitting a `glm` model.
`offset`	this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of observations. One or more offset terms can be included in the formula instead or as well, and if more than one is specified their sum is used.
`control_glm`	optional list of control parameters passed to `glm.control` when computing GLM-based starting values. Ignored when `beta_start` is supplied.
`use_p`	logical indicating whether to apply the `N - p` degrees-of-freedom correction when estimating the scale and working correlation parameters, `p` is the number of regression parameters. Defaults to `TRUE`.
`alpha_vector`	numeric vector of fixed association parameters used only when `corstr = "fixed"`. Must have length `choose(T, 2)` where `T = max(repeated)` after recoding. Ignored otherwise.
`phi_fixed`	logical indicating whether the scale parameter is fixed at the value of `phi_value`. Defaults to `phi_fixed = FALSE`.
`phi_value`	positive number giving the fixed value of the scale parameter. Used only when `phi_fixed = TRUE`. Defaults to `1`.
`...`	additional arguments passed to or from other methods.

Details

`method` specifies the estimation approach. If `method = "gee"`, the standard GEE are solved with no adjustment. If `method` is one of `"brgee-naive"`, `"brgee-robust"` or `"brgee-empirical"`, an adjustment vector is added to produce naive, robust or empirical bias-reduced estimators, respectively. If `method` is one of `"bcgee-naive"`, `"bcgee-robust"` or `"bcgee-empirical"`, the corresponding bias-corrected estimators are produced via a one-step correction applied to the converged GEE solution. If `method = "pgee-jeffreys"`, the GEE are penalized using a Jeffreys-prior penalty run to full convergence. If `method = "opgee-jeffreys"`, a single penalized scoring step is performed from the converged independence penalized solution (one-step approximation). If `method = "hpgee-jeffreys"`, a single standard GEE scoring step is performed from the converged independence penalized solution (hybrid one-step approximation).

For the construction of the `formula` argument, see the documentation of `glm` and `formula`.

The `data` must be in long format (one row per observation). See `reshape` for details on reshaping between long and wide formats.

The `quasi`, `quasibinomial` and `quasipoisson` families are internally remapped to their standard parametric equivalents before fitting: `quasibinomial` to `binomial`, `quasipoisson` to `poisson`, and `quasi` to the family matching its variance function (`gaussian`, `binomial`, `poisson`, `Gamma`, or `inverse.gaussian`). The scale parameter `phi` is then estimated freely from the data unless `phi_fixed = TRUE`.

The default set for the `id` labels is $\{1, \ldots, N\}$, where N is the number of clusters. Otherwise, the function recodes the given labels of `id` onto this set.

The argument `repeated` can be safely omitted only if observations are already ordered within each cluster as intended. If `repeated` is not provided, it is created as `1, 2, ..., n_i` within each cluster i, using the current row order (before internal sorting). If `repeated` is provided, its labels are recoded to `1, ..., T` and must be unique within each cluster.

The variables `id` and `repeated` do not need to be pre-sorted. Instead the function sorts observations in ascending order of `id` and `repeated`.

A term of the form `offset(expression)` is allowed in the right-hand side of `formula`.

The length of `id` and of `repeated` or `weights` (when provided) must equal the number of observations.

Value

An object of class `"geer"`, a list with components:

`coefficients`	a named vector of estimated regression coefficients.
`residuals`	the working residuals.
`fitted.values`	the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.
`rank`	the numeric rank of the fitted model.
`family`	the `family` object used.
`linear.predictors`	the linear fit on the link scale.
`iter`	the number of iterations used.
`prior.weights`	the weights initially supplied, a vector of 1s if none were.
`df.residual`	the residual degrees of freedom.
`y`	the response vector.
`x`	the model matrix.

`qr`	the QR decomposition of the model matrix, used for estimability checking.
`id`	the recoded cluster identifier vector.
`repeated`	the recoded within-cluster ordering vector.
`converged`	logical indicating whether the algorithm converged.
`call`	the matched call.
`formula`	the formula supplied.
`terms`	the `terms` object used.
`data`	the data argument.
`offset`	the offset vector used.
`control`	the `geer_control` list used.
`method`	character string identifying the estimation method used.
`contrasts`	the contrasts used.
`xlevels`	a record of the levels of the factors used in fitting.
`na.action`	information on how missing values were handled, as returned by the `na.action` attribute of the model frame. `NULL` if no observations were removed.
`naive_covariance`	the model-based (naive) covariance matrix.
`robust_covariance`	the sandwich (robust) covariance matrix.
`bias_corrected_covariance`	the bias-corrected covariance matrix.
`association_structure`	the name of the working association structure.
`alpha`	a vector of the estimated working association parameters.
`phi`	the estimated or fixed scale parameter.
`obs_no`	the total number of observations.
`clusters_no`	the number of clusters.
`min_cluster_size`	the minimum cluster size.
`max_cluster_size`	the maximum cluster size.

Note on returned components

For `geewa`, the `alpha` contains the estimated (or fixed) working correlation parameters. The `association_structure` component stores the value of `corstr`. Under `corstr = "independence"`, `alpha` is set to `0`.

For `method` in `"bcgee-naive"`, `"bcgee-robust"`, `"bcgee-empirical"`, `"opgee-jeffreys"`, and `"hpgee-jeffreys"`, `converged` is always `TRUE` in the returned object.

References

Touloumis, A. (2026) Jeffreys-Type Penalized GEE for Correlated Binary Data with an Odds-Ratio Parameterization. *Preprint*. `https://arxiv.org/abs/2606.16058`

Touloumis, A. (2026) Bias-Reduced GEE via Adjusted Estimating Equations, with Odds-Ratio Extensions. *Preprint*. `https://arxiv.org/abs/2606.16043`

See Also

geewa_binary, geer_control, summary.geer, geecriteria.

Examples

```
data("epilepsy", package = "geer")
fitted_model_gee <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  corstr = "exchangeable",
  method = "gee"
)
summary(fitted_model_gee, cov_type = "bias-corrected")

fitted_model_brgee_robust <- update(fitted_model_gee, method = "brgee-robust")
summary(fitted_model_brgee_robust, cov_type = "bias-corrected")

fitted_model_brgee_naive <- update(fitted_model_gee, method = "brgee-naive")
summary(fitted_model_brgee_naive, cov_type = "bias-corrected")

fitted_model_brgee_empirical <- update(fitted_model_gee, method = "brgee-empirical")
summary(fitted_model_brgee_empirical, cov_type = "bias-corrected")

fitted_model_bcgee_robust <- update(fitted_model_gee, method = "bcgee-robust")
summary(fitted_model_bcgee_robust, cov_type = "robust")


## Penalized GEE with custom control
fitted_model_pgee <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  corstr = "exchangeable",
  method = "pgee-jeffreys",
  control = geer_control(jeffreys_power = 0.1)
)
summary(fitted_model_pgee, cov_type = "robust")
```

`geewa_binary` *Fitting (Adjusted) Generalized Estimating Equations for Binary Responses*

Description

Fits a marginal model for repeated or clustered binary responses using Generalized Estimating Equations (GEE). Supported estimation methods include the traditional GEE, bias-reducing GEE, bias-correcting GEE, and Jeffreys-prior penalized GEE.

Usage

```
geewa_binary(
  formula,
  link = "logit",
  data = parent.frame(),
  id,
  repeated,
  control = geer_control(...),
  orstr = "independence",
  method = "gee",
  weights,
  beta_start = NULL,
  offset,
  control_glm = list(...),
  alpha_vector = NULL,
  ...
)
```

Arguments

`formula`	`formula` expression of the form `response` ~ `predictors`: a symbolic description of the marginal model to be fitted.
`link`	character string specifying the link function for the marginal mean model. Options are `"logit"`, `"probit"`, `"cauchit"`, `"cloglog"`, `"identity"`, `"log"`, `"sqrt"`, `"1/mu^2"` and `"inverse"`. Defaults to `"logit"`.
`data`	optional data frame containing variables referenced in `formula`, `id` and `repeated`.
`id`	variable identifying the clusters.
`repeated`	optional variable identifying the order of observations within each cluster.
`control`	a `geer_control` list specifying convergence tolerance, maximum iterations, step-halving parameters, and the Jeffreys-prior power. Defaults to `geer_control()`.
`orstr`	character string specifying the working odds-ratio structure for the within-cluster association. Options are `"independence"`, `"exchangeable"`, `"unstructured"` and `"fixed"`. Defaults to `"independence"`.
`method`	character string specifying the estimation method. Options are the traditional GEE (`"gee"`), bias-reducing methods (`"brgee-robust"`, `"brgee-empirical"`, `"brgee-naive"`), bias-corrected methods (`"bcgee-robust"`, `"bcgee-empirical"`, `"bcgee-naive"`), the fully iterated Jeffreys-prior penalized GEE (`"pgee-jeffreys"`), the one-step penalized GEE (`"opgee-jeffreys"`), and the hybrid one-step GEE (`"hpgee-jeffreys"`). Defaults to `"gee"`.

`weights`	optional numeric vector of observation weights. Must be finite and strictly positive. If not supplied, all weights are 1.
`beta_start`	optional numeric vector of starting values for the regression parameters. If `NULL` (default), starting values are computed by fitting a `glm` model.
`offset`	this can be used to specify an a priori known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector of length equal to the number of observations. One or more offset terms can be included in the formula instead or as well, and if more than one is specified their sum is used.
`control_glm`	optional list of control parameters passed to `glm.control` when computing GLM-based starting values. Ignored when `beta_start` is supplied.
`alpha_vector`	numeric vector of fixed odds-ratio parameters used only when `orstr = "fixed"`. Must have length `choose(T, 2)` where `T = max(repeated)` after recoding, and all elements must be finite and strictly positive. Ignored for all other values of `orstr`.
`...`	additional arguments passed to or from other methods.

Details

`method` specifies the estimation approach. If `method = "gee"`, the standard GEE are solved with no adjustment. If `method` is one of `"brgee-naive"`, `"brgee-robust"` or `"brgee-empirical"`, an adjustment vector is added to produce naive, robust or empirical bias-reducing estimators, respectively. If `method` is one of `"bcgee-naive"`, `"bcgee-robust"` or `"bcgee-empirical"`, the corresponding bias-corrected estimators are produced via a one-step correction applied to the converged GEE solution. If `method = "pgee-jeffreys"`, the GEE are penalized using a Jeffreys-prior penalty run to full convergence. If `method = "opgee-jeffreys"`, a single penalized scoring step is performed from the converged independence penalized solution (one-step approximation). If `method = "hpgee-jeffreys"`, a single standard GEE scoring step is performed from the converged independence penalized solution (hybrid one-step approximation).

The marginal mean model always uses a `binomial` family with the specified `link`. Within-cluster association is modeled through marginalized pairwise odds ratios rather than a correlation structure. The scale parameter is fixed at `phi = 1`.

For the construction of the `formula` argument, see the documentation of `glm` and `formula`.

The `data` must be in long format (one row per observation). See `reshape` for details on reshaping between long and wide formats.

The default set for the `id` labels is $\{1, \dots, N\}$, where N is the number of clusters. Otherwise, the function recodes the given labels of `id` onto this set.

The argument `repeated` can be safely omitted only if observations are already ordered within each cluster as intended. If `repeated` is not provided, it is created as `1, 2, ..., n_i` within each cluster i, using the current row order (before internal sorting). If `repeated` is provided, its labels are recoded to `1, ..., T` and must be unique within each cluster.

The variables `id` and `repeated` do not need to be pre-sorted. Instead the function sorts observations in ascending order of `id` and `repeated`.

A term of the form `offset(expression)` is allowed in the right-hand side of `formula`.

The length of `id` must equal the number of observations. When provided, `repeated` and `weights` must also have the same length.

Value

An object of class `"geer"`, a list with components:

`coefficients`	a named vector of estimated regression coefficients.
`residuals`	the working residuals.
`fitted.values`	the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.
`rank`	the numeric rank of the fitted model.
`family`	the `family` object used.
`linear.predictors`	the linear fit on the link scale.
`iter`	the number of iterations used.
`prior.weights`	the weights initially supplied, a vector of 1s if none were.
`df.residual`	the residual degrees of freedom.
`y`	the response vector.
`x`	the model matrix.
`qr`	the QR decomposition of the model matrix, used for estimability checking.
`id`	the recoded cluster identifier vector.
`repeated`	the recoded within-cluster ordering vector.
`converged`	logical indicating whether the algorithm converged.
`call`	the matched call.
`formula`	the formula supplied.
`terms`	the `terms` object used.
`data`	the data argument.
`offset`	the offset vector used.
`control`	the `geer_control` list used.
`method`	character string identifying the estimation method used.
`contrasts`	the contrasts used.
`xlevels`	a record of the levels of the factors used in fitting.
`na.action`	information on how missing values were handled, as returned by the `na.action` attribute of the model frame. `NULL` if no observations were removed.
`naive_covariance`	the model-based (naive) covariance matrix.
`robust_covariance`	the sandwich (robust) covariance matrix.
`bias_corrected_covariance`	the bias-corrected covariance matrix.
`association_structure`	the name of the working association structure.
`alpha`	a vector of the estimated working association parameters.

`phi`	the estimated or fixed scale parameter.
`obs_no`	the total number of observations.
`clusters_no`	the number of clusters.
`min_cluster_size`	the minimum cluster size.
`max_cluster_size`	the maximum cluster size.

Note on returned components

For `geewa_binary`, the `phi` component is always `1` and `alpha` contains the estimated (or fixed) working odds-ratio parameters, not correlation parameters. The `association_structure` component stores the value of `orstr`. Under `orstr = "independence"`, `alpha` is set to `1`.

For `method` in `"bcgee-naive"`, `"bcgee-robust"`, `"bcgee-empirical"`, `"opgee-jeffreys"`, and `"hpgee-jeffreys"`, `converged` is always `TRUE` in the returned object, because these methods produce their estimate via a single correction step applied to an already-converged fit.

References

Touloumis, A. (2026) Jeffreys-Type Penalized GEE for Correlated Binary Data with an Odds-Ratio Parameterization. *Preprint*. `https://arxiv.org/abs/2606.16058`

Touloumis, A. (2026) Bias-Reduced GEE via Adjusted Estimating Equations, with Odds-Ratio Extensions. *Preprint*. `https://arxiv.org/abs/2606.16043`

See Also

`geewa`, `geer_control`, `summary.geer`, `geecriteria`.

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", , drop = FALSE]
fitted_model <- geewa_binary(
  formula = status ~ baseline + treatment * gender + visit * age,
  link = "probit",
  data = respiratory2,
  id = id,
  repeated = visit,
  orstr = "independence",
  method = "pgee-jeffreys"
)
summary(fitted_model, cov_type = "bias-corrected")


data("cholecystectomy", package = "geer")
fitted_model_gee <- geewa_binary(
  formula = pain ~ treatment + gender + age,
  link = "logit",
  data = cholecystectomy,
  id = id,
```

```
  orstr = "independence",
  method = "gee"
)
summary(fitted_model_gee, cov_type = "bias-corrected")

fitted_model_brgee_robust <- update(fitted_model_gee, method = "brgee-robust")
summary(fitted_model_brgee_robust, cov_type = "bias-corrected")

fitted_model_brgee_naive <- update(fitted_model_gee, method = "brgee-naive")
summary(fitted_model_brgee_naive, cov_type = "bias-corrected")

fitted_model_brgee_empirical <- update(fitted_model_gee, method = "brgee-empirical")
summary(fitted_model_brgee_empirical, cov_type = "bias-corrected")

fitted_model_bcgee_robust <- update(fitted_model_gee, method = "bcgee-robust")
summary(fitted_model_bcgee_robust, cov_type = "robust")


## Penalized GEE with exchangeable odds-ratio structure
fitted_model_pgee <- geewa_binary(
  formula = pain ~ treatment + gender + age,
  link = "logit",
  data = cholecystectomy,
  id = id,
  orstr = "exchangeable",
  method = "pgee-jeffreys",
  control = geer_control(jeffreys_power = 1)
)
summary(fitted_model_pgee, cov_type = "robust")
```

`glance.geer` *Glance at a geer Object*

Description

Produces a one-row summary for a fitted `geer` object, following **broom** conventions.

Usage

```
## S3 method for class 'geer'
glance(x, ...)
```

Arguments

`x`	an object of class "`geer`".
`...`	additional arguments passed to or from other methods. Currently unused.

Details

The returned one-row data frame contains the following columns:

`family` name of the marginal response family.

`link` name of the link function.

`method` estimation method, for example `"gee"`, `"brgee-robust"`, `"pgee-jeffreys"`, `"opgee-jeffreys"`, or `"hpgee-jeffreys"`.

`wastr` stored working association structure. For `geewa()` fits this corresponds to `corstr`; for `geewa_binary()` fits it corresponds to `orstr`.

`nobs` total number of observations $n^\star$.

`nclusters` number of independent clusters N.

`min.cluster.size` minimum cluster size.

`max.cluster.size` maximum cluster size.

`npar` number of marginal mean-model parameters p.

`df.residual` residual degrees of freedom $n^\star - p$.

`phi` estimated or fixed dispersion parameter. This equals `1` for the odds-ratio parameterization used by `geewa_binary()`.

`QIC` Quasi Information Criterion (Pan, 2001), computed from the robust sandwich covariance estimator. Smaller values are preferred.

`QICu` Covariate-selection variant of QIC (Pan, 2001). Smaller values are preferred.

`CIC` Correlation Information Criterion (Hin and Wang, 2009), used here to compare working association structures. Smaller values are preferred.

`converged` logical; `TRUE` if the fitting algorithm converged.

`niter` number of iterations used.

QIC and CIC are computed using the same formulas as `geecriteria(object, cov_type = "robust")`. QICu does not depend on the covariance estimator. If computation fails, the corresponding values are returned as `NA_real_`. For the full set of model selection criteria, including RJC, GESSC, and GPC, see `geecriteria`.

Value

A one-row data frame with columns as described in the Details section. When **tibble** is available, the result has class `c("tbl_df", "tbl", "data.frame")`. Otherwise, a plain `data.frame` is returned.

References

Pan W. (2001) Akaike's information criterion in generalized estimating equations. *Biometrics*, **57**, 120–125.

Hin L.Y. and Wang Y.G. (2009) Working-correlation-structure identification in generalized estimating equations. *Statistics in Medicine*, **28**, 642–658.

Robinson D., Hayes A. and Couch S. (2024) *broom: Convert Statistical Objects into Tidy Tibbles*. `https://broom.tidymodels.org/`.

See Also

`tidy.geer`, `geecriteria`, `geewa`, `geewa_binary`.

Examples

```
data("epilepsy", package = "geer")
fitmodel <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  corstr = "exchangeable",
  method = "gee"
)
glance(fitmodel)

data("cerebrovascular", package = "geer")
fitbin <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
glance(fitbin)


fitind  <- update(fitmodel, corstr = "independence")
fitar1  <- update(fitmodel, corstr = "ar1")
fitunst <- update(fitmodel, corstr = "unstructured")
do.call(rbind, lapply(
  list(independence = fitind, exchangeable = fitmodel, ar1 = fitar1,
       unstructured = fitunst),
  glance
))[, c("wastr", "QIC", "CIC", "niter")]
```

`leprosy` *Antibiotic Treatment for Leprosy Trial*

Description

Data from a randomized clinical trial on the efficacy of antibiotic treatments for leprosy in the Philippines.

Usage

```
leprosy
```

Format

A data frame with 60 rows and 4 columns:

id integer subject identifier.

period factor period identifier with levels `pre` and `post`.

bacilli integer count of leprosy bacilli at six body sites.

treatment factor with levels `A`, `B`, and `C`, where `C` denotes placebo.

Details

Thirty subjects in the Philippines were enrolled in a randomized clinical trial, yielding 60 observations (two periods per subject). Subjects were assigned to receive treatment A, B, or C (placebo). Before treatment, the number of leprosy bacilli at six body sites was recorded. After treatment, bacilli counts were recorded again. The trial aimed to assess whether treatments A and B reduced bacilli abundance compared with placebo.

References

Snedecor, G.W. and Cochran, W.G. (1967) *Statistical Methods*. Ames, Iowa: Iowa State University Press.

Examples

```
data("leprosy", package = "geer")
str(leprosy)
```

`model.matrix.geer` *Construct Design Matrices from a geer Object*

Description

Constructs the design matrix for the marginal mean model underlying a fitted `geer` object.

Usage

```
## S3 method for class 'geer'
model.matrix(object, ...)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`...`	additional arguments passed to or from other methods.

Details

The design matrix is reconstructed from `object$terms` and `object$data`, with factor variables and interactions expanded using the contrasts recorded at fitting time.

By convention, if the response variable also appears on the right-hand side of the formula, it is dropped, although interactions involving that term are retained.

Value

A numeric design matrix for the marginal mean model represented by `object`.

The returned matrix has an `"assign"` attribute: an integer vector with one entry per column giving the index of the formula term that produced that column. A value of `0` corresponds to the intercept, if present. Positive values index terms in the order given by `attr(terms(object), "term.labels")`.

If the model contains factor terms, the returned matrix may also have a `"contrasts"` attribute giving the contrasts used for those factors.

See Also

`geewa`, `geewa_binary`, `model.matrix`.

Examples

```
data("leprosy", package = "geer")
fit <- geewa(
  formula = bacilli ~ factor(period) + factor(period):treatment,
  family = poisson(link = "log"),
  data = leprosy,
  id = id
)
model.matrix(fit)

data("cerebrovascular", package = "geer")
fit <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id
)
model.matrix(fit)
```

`predict.geer` *Predictions from a geer Object*

Description

Generates predictions from a fitted `geer` object, optionally with approximate standard errors.

Usage

```
## S3 method for class 'geer'
predict(
  object,
  newdata = NULL,
  type = c("link", "response"),
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  se.fit = FALSE,
  ...
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`newdata`	optional data frame in which to look for variables used for prediction. If omitted, predictions are made on the data used for fitting.
`type`	character string specifying the scale of the predictions. Options are `"link"` for the linear predictor scale and `"response"` for the response scale. Defaults to `"link"`.
`cov_type`	character string specifying the covariance matrix estimator used to compute approximate standard errors when `se.fit = TRUE`. Options are the bias-corrected estimator (`"bias-corrected"`), the sandwich or robust estimator (`"robust"`), the degrees-of-freedom adjusted estimator (`"df-adjusted"`), and the model-based or naive estimator (`"naive"`). Defaults to `"bias-corrected"`.
`se.fit`	logical indicating whether approximate standard errors are to be returned. Defaults to `FALSE`.
`...`	additional arguments passed to or from other methods.

Details

Predictions are obtained by computing the model matrix for `newdata` (or using the original fit when `newdata` is omitted) and multiplying by the estimated coefficients. If `type = "response"`, the linear predictor is transformed via the inverse link function.

When `se.fit = TRUE`, approximate standard errors are computed by the delta method using the covariance matrix specified by `cov_type`. On the response scale, standard errors are additionally scaled by the absolute derivative of the inverse link function.

Value

If `se.fit = FALSE`, a numeric vector of predictions on the scale specified by `type`. If `se.fit = TRUE`, a list with components:

`fit`	numeric vector of predictions.
`se.fit`	numeric vector of approximate standard errors of the predictions.

See Also

`fitted.geer`, `residuals.geer`, `geewa`, `geewa_binary`, `predict`.

Examples

```
data("cerebrovascular", package = "geer")
fit <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
head(predict(fit, type = "link"))
head(predict(fit, type = "response"))

nd <- cerebrovascular[1:5, , drop = FALSE]
predict(fit, newdata = nd, type = "response")

pred <- predict(fit, type = "response", se.fit = TRUE, cov_type = "robust")
head(pred$fit)
head(pred$se.fit)
```

`print.geer` *Print a geer Object*

Description

Prints the call, estimated regression coefficients, and basic fitting information for a fitted `geer` object.

Usage

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

Arguments

`x`	a fitted model object of class `"geer"`.
`...`	additional arguments passed to or from other methods.

Value

The input object `x` is returned invisibly.

See Also

`summary.geer`, `coef.geer`.

Examples

```
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  corstr = "exchangeable"
)
print(fit)
```

`print.summary.geer` *Print a summary.geer Object*

Description

Prints the contents of a `summary.geer` object, including the call, estimation method, coefficient table, dispersion parameter, and working association structure.

Usage

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

Arguments

`x`	an object of class `"summary.geer"`.
`...`	additional arguments passed to or from other methods. Currently unused.

Value

The input object `x` is returned invisibly.

See Also

`summary.geer`, `print.geer`.

Examples

```
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  corstr = "exchangeable"
)
print(summary(fit))
```

---

residuals.geer *Residuals from a geer Object*

Description

Extracts residuals of different types from a fitted `geer` object.

Usage

```
## S3 method for class 'geer'
residuals(object, type = c("working", "pearson", "deviance"), ...)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`type`	character string specifying the type of residuals to return. Options are `"working"` for raw residuals, `"pearson"` for residuals standardized by the variance function, and `"deviance"` for signed square roots of the deviance contributions. Defaults to `"working"`.
`...`	additional arguments passed to or from other methods.

Details

Residuals are computed using the marginal variance and deviance functions of the family specified in the fitted model.

Value

A numeric vector of residuals of the requested `type`, of the same length as the number of observations used in fitting.

See Also

`fitted.geer`, `predict.geer`, `geewa`, `geewa_binary`, `residuals`.

Examples

```
data("cerebrovascular", package = "geer")
fit <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
head(residuals(fit, type = "working"))
head(residuals(fit, type = "pearson"))
head(residuals(fit, type = "deviance"))
```

`respiratory` *Respiratory Illness Clinical Trial*

Description

Data from a randomized clinical trial on respiratory illness.

Usage

```
respiratory
```

Format

A data frame with 444 rows and 8 columns:

id integer subject identifier.

visit integer follow-up visit identifier.

status binary indicator of respiratory status, 0 = poor and 1 = good.

treatment factor with levels `active` and `placebo`.

baseline binary indicator of baseline respiratory status, 0 = poor and 1 = good.

age numeric age in years recorded at baseline.

gender factor with levels `female` and `male`.

center factor with levels `C1` and `C2`.

Details

One hundred eleven subjects from two clinical centers were enrolled in a randomized clinical trial, yielding 444 observations (four follow-up visits per subject). Subjects were randomly assigned to treatment groups. Baseline examinations were conducted before treatment began. After treatment, subjects were examined at four scheduled follow-up visits.

References

Stokes, M.E., Davis, C.S. and Koch, G.G. (1995) *Categorical Data Analysis using the SAS System*. Cary, NC: SAS Institute, Inc.

Examples

```
data("respiratory", package = "geer")
str(respiratory)
```

`rinse` *Dental Plaque Mouth Rinse Trial*

Description

Data from a randomized clinical trial on the efficacy of mouth rinses in reducing dental plaque.

Usage

```
rinse
```

Format

A data frame with 218 rows and 8 columns:

id integer subject identifier.

time integer follow-up month identifier.

score numeric plaque score.

treatment factor indicating rinse group with levels `A`, `B`, and `placebo`.

baseline numeric plaque score at baseline.

gender factor with levels `female` and `male`.

age numeric age in years recorded at baseline.

smoke factor with levels `yes` and `no`.

Details

One hundred nine adults aged 18–55 with pre-existing dental plaque but without advanced periodontal disease were enrolled in a randomized, double-blind clinical trial, yielding 218 observations (baseline plus up to two follow-up assessments per subject). Subjects were assigned to one of two novel mouth rinses (A or B) or to a control mouth rinse. Eligibility required at least 20 sound natural teeth and a mean plaque index of 2.0 or greater. Plaque was assessed at baseline, 3 months, and 6 months using the Turesky modification of the Quigley-Hein index. Four subjects had missing plaque scores. The trial aimed to evaluate the effectiveness of the three rinses in inhibiting dental plaque.

References

Hadgu, A. and Koch, G. (1999) Application of generalized estimating equations to a dental randomized clinical trial. *Journal of Biopharmaceutical Statistics*, **9**, 161–178.

Examples

```
data("rinse", package = "geer")
str(rinse)
```

`step_p` *Stepwise Model Selection by Hypothesis Testing*

Description

Performs stepwise model selection for a fitted `geer` object using repeated single-term additions (`add1.geer`) and deletions (`drop1.geer`), with candidate moves evaluated by hypothesis tests.

Usage

```
step_p(
  object,
  scope,
  direction = c("backward", "forward", "both"),
  p_enter = 0.15,
  p_remove = 0.15,
  test = c("wald", "score", "working-wald", "working-score", "working-lrt"),
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  pmethod = c("rao-scott", "satterthwaite"),
  steps = 1000
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`scope`	defines the range of models examined in the stepwise search. This should be either a single formula, or a list containing components `upper` and `lower`, both formulae. See the details for how to specify the formulae and how they are used.
`direction`	character string specifying the direction of the stepwise search: backward elimination (`"backward"`), forward selection (`"forward"`) or bidirectional (`"both"`). Defaults to `"backward"`.
`p_enter`	numeric value specifying the p-value threshold for adding a term during forward steps. Defaults to `0.15`.
`p_remove`	numeric value specifying the p-value threshold for removing a term during backward steps. Defaults to `0.15`.
`test`	character string specifying the hypothesis testing procedure. Options are the Wald test (`"wald"`), the generalized score test (`"score"`), the modified working Wald test (`"working-wald"`), the modified working score test (`"working-score"`), and the modified working likelihood ratio test (`"working-lrt"`). Defaults to `"wald"`.
`cov_type`	character string specifying the covariance matrix estimator used for inference on the regression parameters. Options are the bias-corrected estimator (`"bias-corrected"`), the sandwich or robust estimator (`"robust"`), the degrees-of-freedom adjusted estimator (`"df-adjusted"`), and the model-based or naive estimator (`"naive"`). Defaults to `"bias-corrected"`.

`pmethod`	character string specifying the approximation used to compute the p-value for the modified working tests. Options are the Rao–Scott approximation (`"rao-scott"`) and the Satterthwaite approximation (`"satterthwaite"`). Defaults to `"rao-scott"`.
`steps`	non-negative integer giving the maximum number of accepted steps to perform. If `steps = 0`, the initial model is returned with an initial-model row in its `anova` component. Defaults to `1000`.

Details

`step_p()` repeatedly calls `add1.geer` and `drop1.geer`.

The set of models searched is determined by the `scope` argument. The right-hand side of its `lower` component is always retained in the model, and the right-hand side of its `upper` component defines the largest model that can be considered. If `scope` is a single formula, it specifies the `upper` component and the `lower` model is empty.

Candidate additions and deletions respect model hierarchy: a move is allowed only if it preserves all lower-order component terms implied by any higher-order interactions in the model.

If `scope` is omitted, the search is restricted to the terms in the initial model.

With `direction = "both"`, backward elimination is attempted first at each step. A forward addition is considered only if no eligible backward deletion is found. The algorithm stops when no candidate move satisfies the relevant p-value threshold, when the maximum number of steps is reached, or when an immediate add-remove cycle involving the same term would occur.

Details of the hypothesis tests controlled by `test` are given in Rotnitzky and Jewell (1990). The option `test = "working-lrt"` is valid only when the model is fitted with an independence working association structure. Otherwise, an error is returned.

When `test %in% c("wald", "score")`, the `pmethod` argument is ignored and `cov_type` specifies the covariance estimator used to compute the test statistic. For modified working tests, `cov_type` determines the covariance matrix used to form the coefficients of the sum of independent chi-squared random variables, and `pmethod` specifies the approximation used to compute the p-value.

Value

A fitted model object of class `"geer"` corresponding to the final selected model. The returned object also includes an `anova` component of class `c("anova", "data.frame")` summarizing the stepwise sequence. This table has one row for the initial model and one row for each accepted step, with columns `Step` (the term added or removed), `Df` (degrees of freedom of the test), `Chi` (test statistic), `Pr(>Chi)` (p-value), and `CIC` (Correlation Information Criterion).

References

Rotnitzky, A. and Jewell, N.P. (1990) Hypothesis testing of regression parameters in semiparametric generalized linear models for cluster correlated data. *Biometrika*, **77**, 485–497.

See Also

`add1.geer`, `drop1.geer`, `anova.geer`, `geecriteria`, `geewa`, `geewa_binary`.

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", , drop = FALSE]

full_fit <- geewa_binary(
  formula = status ~ (baseline + treatment + gender + visit + age)^2,
  link = "probit",
  data = respiratory2,
  id = id,
  repeated = visit,
  orstr = "independence",
  method = "pgee-jeffreys"
)

## Backward elimination within the initial model terms
step_p(
  full_fit,
  direction = "backward",
  test = "wald",
  cov_type = "bias-corrected",
  p_remove = 0.10
)

## Bidirectional selection with an explicit scope
step_p(
  full_fit,
  scope = list(
    lower = ~ baseline + treatment,
    upper = ~ (baseline + treatment + gender + visit + age)^2
  ),
  direction = "both",
  test = "score",
  cov_type = "robust",
  p_enter = 0.10,
  p_remove = 0.15,
  steps = 50
)
```

`summary.geer` *Summarize a geer Object*

Description

Produces a coefficient table and basic model summary information for a fitted `geer` object.

Usage

```
## S3 method for class 'geer'
summary(
  object,
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  ...
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`cov_type`	character string specifying the covariance estimator used to compute standard errors, z-statistics, and p-values. Options are `"bias-corrected"`, `"robust"`, `"df-adjusted"`, and `"naive"`. Defaults to `"bias-corrected"`.
`...`	additional arguments passed to or from other methods. Currently unused.

Value

An object of class `"summary.geer"`, a list with components:

`coefficients`	a matrix with columns `Estimate`, `Std. Error`, `z` value, and `Pr(>|z|)`, with one row per regression parameter.
`family`	the `family` object used.
`alpha`	the estimated or fixed association parameters.
`call`	the matched call.
`residuals`	the working residuals.
`iter`	the number of iterations used.
`converged`	logical indicating whether the algorithm converged.
`phi`	the estimated or fixed scale parameter.
`association_structure`	the name of the working association structure.
`method`	character string identifying the estimation method used.
`cov_type`	the covariance estimator used for standard errors.

See Also

`print.summary.geer`, `tidy.geer`, `glance.geer`, `vcov.geer`.

Examples

```
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
```

```
  corstr = "exchangeable"
)
summary(fit)
summary(fit, cov_type = "bias-corrected")

data("cerebrovascular", package = "geer")
fit2 <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
summary(fit2)
```

`tidy.geer` *Tidy a geer Object*

Description

Summarizes a fitted `geer` object at the coefficient level, returning one row per regression term in a tidy data frame.

Usage

```
## S3 method for class 'geer'
tidy(
  x,
  conf.int = FALSE,
  conf.level = 0.95,
  exponentiate = FALSE,
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  ...
)
```

Arguments

`x`	an object of class `"geer"`.
`conf.int`	logical indicating whether to append confidence interval columns `conf.low` and `conf.high`. Defaults to `FALSE`.
`conf.level`	numeric coverage probability for the confidence interval when `conf.int = TRUE`. Defaults to `0.95`.
`exponentiate`	logical indicating whether to exponentiate coefficient estimates and confidence limits. This is often useful for models with a log, logit, or complementary log-log link. Standard errors and Wald z-statistics are *not* transformed. Defaults to `FALSE`.

`cov_type`	character string specifying the covariance estimator used to compute standard errors and Wald z-statistics. Options are `"bias-corrected"` (default), `"robust"`, `"df-adjusted"`, and `"naive"` (model-based). See `vcov.geer` for details.
`...`	additional arguments passed to or from other methods. Currently unused.

Details

The returned data frame contains the following columns:

`term` name of the regression coefficient.

`estimate` point estimate, or the exponentiated point estimate when `exponentiate = TRUE`.

`std.error` standard error derived from `vcov(x, cov_type = cov_type)`.

`statistic` Wald z-statistic, computed as the coefficient estimate divided by its standard error. `NA` is reported when the standard error is not positive.

`p.value` two-sided p-value from the standard normal distribution.

`conf.low`**,** `conf.high` lower and upper Wald confidence limits. These columns are included only when `conf.int = TRUE`.

When `exponentiate = TRUE`, only `estimate`, `conf.low`, and `conf.high` are exponentiated. The columns `std.error` and `statistic` remain on the original scale.

Value

A data frame with one row per regression coefficient and columns as described in the Details section. When **tibble** is available, the result has class `c("tbl_df", "tbl", "data.frame")`. Otherwise, a plain `data.frame` is returned.

References

Robinson D., Hayes A. and Couch S. (2024) *broom: Convert Statistical Objects into Tidy Tibbles*. `https://broom.tidymodels.org/`.

See Also

`glance.geer`, `vcov.geer`, `confint.geer`, `geewa`, `geewa_binary`.

Examples

```
data("epilepsy", package = "geer")
fitmodel <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  corstr = "exchangeable",
  method = "gee"
)
tidy(fitmodel)
tidy(fitmodel, conf.int = TRUE)
```

```
tidy(fitmodel, conf.int = TRUE, exponentiate = TRUE)
tidy(fitmodel, cov_type = "robust")

data("cerebrovascular", package = "geer")
fitbin <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
tidy(fitbin, conf.int = TRUE, conf.level = 0.90, exponentiate = TRUE)
```

`vcov.geer` *Extract Variance-Covariance Matrix from a geer Object*

Description

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

Usage

```
## S3 method for class 'geer'
vcov(
  object,
  cov_type = c("bias-corrected", "robust", "df-adjusted", "naive"),
  ...
)
```

Arguments

`object`	a fitted model object of class `"geer"`.
`cov_type`	character string specifying the covariance matrix estimator used for inference on the regression parameters. Options are the bias-corrected estimator (`"bias-corrected"`), the sandwich or robust estimator (`"robust"`), the degrees-of-freedom adjusted estimator (`"df-adjusted"`), and the model-based or naive estimator (`"naive"`). Defaults to `"bias-corrected"`.
`...`	additional arguments passed to or from other methods.

Details

The form of the covariance estimator is controlled by `cov_type`:

`"bias-corrected"` the bias-corrected covariance estimator (Morel et al., 2003).

`"robust"` the sandwich (robust) covariance estimator (Liang and Zeger, 1986).

`"df-adjusted"` the degrees-of-freedom adjusted covariance estimator (MacKinnon, 1985).

`"naive"` the model-based covariance estimator (Liang and Zeger, 1986).

Value

A square numeric matrix of estimated covariances between regression coefficients. Rows and columns are named according to the coefficient names returned by `coef.geer`.

References

Liang, K.Y. and Zeger, S.L. (1986) Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 13–22.

MacKinnon, J.G. (1985) Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, **29**, 305–325.

Morel, J.G., Bokossa, M.C. and Neerchal, N.K. (2003) Small sample correction for the variance of GEE estimators. *Biometrical Journal*, **45**, 395–409.

See Also

`coef.geer`, `confint.geer`, `summary.geer`, `tidy.geer`.

Examples

```
data("cerebrovascular", package = "geer")
fit <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  orstr = "exchangeable"
)
vcov(fit)
vcov(fit, cov_type = "robust")
```

Index