

Package: geer (via r-universe)

October 22, 2026

Type Package

Title Bias-Reduced and Penalized Generalized Estimating Equations

Version 0.1.1

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>](https://doi.org/10.48550/arXiv.2606.16043) and Touloumis (2026b) [<doi:10.48550/arXiv.2606.16058>](https://doi.org/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, grDevices, graphics, Rcpp (>= 1.0.9), stats, utils

SystemRequirements C++17

LinkingTo Rcpp, RcppArmadillo

Suggests covr, emmeans (>= 2.0.0), estimability, insight (>= 1.3.0), margineffects (>= 0.32.0), pkgload, testthat (>= 3.0.0), tibble, withr

Encoding UTF-8

LazyData true

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

NeedsCompilation yes

Repository <https://anestistouloumis.r-universe.dev>

Date/Publication 2026-10-21 23:47:58 UTC

RemoteUrl <https://github.com/anestistouloumis/geer>

RemoteRef HEAD

RemoteSha 34624435244b136f49a521e14e6fc9f5899da7d0

Contents

geer-package	3
add1.geer	3
anova.geer	5
cerebrovascular	7
cholecystectomy	8
coef.geer	9
confint.geer	10
depression	11
emmeans-support	12
epilepsy	14
fitted.geer	15
frechet_bounds_cor	16
geecriteria	17
geer_control	21
geewa	22
geewa_binary	27
glance.geer	31
gof_residual_test	33
gof_score_test	35
gof_tests	38
hatvalues.geer	40
leprosy	43
local_influence	44
marginaleffects-support	46
mcar_homoscedasticity_test	47
mcar_little_test	51
mcar_logistic_test	54
model.matrix.geer	57
plot.geer_runs_test	58
predict.geer	60
print.geer	61
print.summary.geer	62
residuals.geer	63
respiratory	65
rinse	66
runs_test	67
step_criterion	69
step_p	72
summary.geer	74
tidy.geer	76
vcov.geer	77

Index

81

Description

Fits marginal models for clustered or repeated responses by generalized estimating equations (GEE; Liang and Zeger, 1986), using ordinary, bias-reduced, or bias-corrected GEE (Touloumis, 2026a) or Jeffreys-type penalized GEE (Touloumis, 2026b). [geewa](#) handles continuous, binary, and count responses with a working correlation structure, and [geewa_binary](#) handles binary responses with a working odds-ratio structure.

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. (2026a) Bias-reduced GEE via adjusted estimating equations, with odds-ratio extensions. *arXiv preprint*. doi:10.48550/arXiv.2606.16043

Touloumis, A. (2026b) Jeffreys-type penalized GEE for correlated binary data with an odds-ratio parameterization. *arXiv preprint*. doi:10.48550/arXiv.2606.16058

See Also

- Model fitting: [geewa](#), [geewa_binary](#), [geer_control](#).
- Summaries and inference: [summary.geer](#), [vcov.geer](#), [confint.geer](#), [predict.geer](#), [tidy.geer](#), [glance.geer](#).
- Model building: [anova.geer](#), [add1.geer](#), [step_p](#), [step_criterion](#), [geecriteria](#).
- Diagnostics: [residuals.geer](#), [hatvalues.geer](#), [local_influence](#), [gof_tests](#), [gof_score_test](#), [gof_residual_test](#), [runs_test](#), [frechet_bounds_cor](#), [mcar_little_test](#), [mcar_logistic_test](#), [mcar_homoscedasticity_test](#).
- Other packages: [emmeans-support](#), [marginaleffects-support](#).

Description

Fits all models that differ from a fitted geer object by adding or dropping a single term in scope, and tests each change.

Usage

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

Arguments

<code>object</code>	a fitted model object of class "geer".
<code>scope</code>	a formula giving the terms to be considered for adding or dropping.
<code>test</code>	character string specifying the hypothesis test: Wald ("wald"), generalized score ("score"), modified working Wald ("working-wald"), modified working score ("working-score"), or modified working likelihood ratio ("working-lrt"). Defaults to "wald".
<code>cov_type</code>	character string specifying the covariance matrix estimator used for inference: one of the estimators described in vcov.geer . Defaults to "bias-corrected".
<code>pmethod</code>	character string specifying the approximation to the null distribution of the modified working tests: Rao-Scott ("rao-scott") or Satterthwaite ("satterthwaite"). Ignored for the Wald and score tests. Defaults to "rao-scott".
<code>...</code>	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Details

For `add1()`, `scope` must contain at least one term not in the model, and `.` denotes the terms already in the model. Additions and deletions respect marginality: a term is not dropped while a higher-order interaction containing it remains, and an interaction is not added without its lower-order terms. The tests are described in [anova.geer](#). The table also reports the correlation information criterion (CIC) of each model, computed with `cov_type`; with `cov_type = "jackknife"` every candidate model is therefore refitted once per cluster.

When `object` was fitted with a data frame, every candidate model is refitted to the rows of that data frame used by `object`, so rows omitted because of missing values stay omitted even if the candidate

model does not use the variable concerned. For `add1()`, a term with missing values in these rows is an error.

Value

An object of class `c("geer_anova", "anova", "data.frame")` with one row for the current model and one for each admissible addition or deletion, and columns `Df`, `CIC`, `Chi`, and `Pr(>Chi)`.

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", ]
fit <- geewa(
  formula = status ~ baseline + treatment + gender + visit + age,
  family = binomial(link = "probit"),
  data = respiratory2,
  id = id,
  repeated = visit,
  corstr = "ar1"
)
## Single-term deletions
drop1(fit, test = "score")

## Single-term additions
add1(fit, scope = ~ . + baseline:age + visit:age + treatment:age,
     test = "score")
```

Description

Computes tables of hypothesis tests for one or more fitted `geer` objects, analogous to analysis of deviance tables for generalized linear models.

Usage

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

Arguments

<code>object</code>	a fitted model object of class "geer".
<code>...</code>	additional fitted model objects of class "geer" to be included in the comparison, supplied without names. Any other argument is an error.
<code>test</code>	character string specifying the hypothesis test: Wald ("wald"), generalized score ("score"), modified working Wald ("working-wald"), modified working score ("working-score"), or modified working likelihood ratio ("working-lrt"). Defaults to "wald".
<code>cov_type</code>	character string specifying the covariance matrix estimator used for inference: one of the estimators described in vcov.geer . Defaults to "bias-corrected".
<code>pmethod</code>	character string specifying the approximation to the null distribution of the modified working tests: Rao-Scott ("rao-scott") or Satterthwaite ("satterthwaite"). Ignored for the Wald and score tests. Defaults to "rao-scott".

Details

The tests are those of Rotnitzky and Jewell (1990). The Wald and score statistics use the covariance matrix selected by `cov_type`. The modified working tests compare a working statistic with a weighted sum of independent chi-squared variables, whose weights depend on `cov_type`, and `pmethod` selects the approximation to its distribution. The modified working likelihood ratio test requires a working independence structure, and both working log-likelihoods are scaled by the dispersion of the larger model. The score tests are evaluated at the fitted means of the smaller model with the association and dispersion parameters of the larger model; with `cov_type = "jackknife"`, the covariance component is the jackknife covariance of the larger model. The small-sample estimators of [vcov.geer](#), such as "kauermann-carroll", are evaluated at the same fitted means, which extends their published definitions.

With several models, the fits must use the same data and each consecutive pair must be nested. A model whose working association structure differs from that of the first model is omitted with a warning, and any other difference in the fitting settings, such as the estimation method or the handling of the dispersion, is an error.

With one model, the sequential models are refitted to the rows used by the fit, as described in [add1.geer](#).

Value

An object of class `c("geer_anova", "anova", "data.frame")` with columns `Df`, `Resid. Df`, `Chi`, and `Pr(>Chi)`. With one model, the terms are tested sequentially, from first to last; with several models, each model is compared with the previous one. `Resid. Df` is the number of observations minus the number of regression parameters.

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](#), [step_p](#), [geecriteria](#), [geewa](#), [geewa_binary](#).

Examples

```
data("cerebrovascular", package = "geer")
fit <- geewa_binary(
  formula = ecg ~ treatment + factor(period),
  link = "logit",
  data = cerebrovascular,
  id = id,
  repeated = period,
  orstr = "exchangeable"
)
## Sequential tests of the terms
anova(fit)

## Comparison of nested models
fit_null <- update(fit, . ~ . - treatment)
anova(fit_null, fit, test = "score")
```

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 placebo and active.

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.

Source

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

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 placebo and active.

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 (Jorgensen et al., 1995), 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

<code>object</code>	a fitted model object of class "geer".
<code>...</code>	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

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

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 = "bias-corrected", ...)
```

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	a single number strictly between 0 and 1 specifying the confidence level. Defaults to 0.95.
cov_type	character string specifying the covariance matrix estimator used for the standard errors: one of the estimators described in vcov.geer . Defaults to "bias-corrected".
...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Details

The intervals are $\hat{\beta} \pm z_{1-\alpha/2} \text{SE}(\hat{\beta})$, with standard errors from `vcov(object, cov_type = cov_type)`.

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,
  repeated = period,
  orstr = "exchangeable"
)
confint(fit)
confint(fit, parm = "treatmentactive", cov_type = "robust")
```

depression	<i>Postnatal Depression Oestrogen Patch Trial</i>
------------	---

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 (Gregoire et al., 1996), 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 **emmeans** functions, such as `emmeans::emmeans()`, to be applied to fitted `geer` objects. They are called by **emmeans** and not usually by the user. Inference is asymptotic, with infinite degrees of freedom, and uses the bias-corrected covariance matrix unless `vcov.method` requests another estimator.

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

<code>object</code>	a fitted model object of class "geer".
<code>data</code>	optional data frame from which emmeans recovers the predictors. Defaults to NULL, which uses the data used to fit the model.
<code>...</code>	additional arguments passed through from emmeans .
<code>trms</code>	the terms component supplied by emmeans .
<code>xlev</code>	the factor levels supplied by emmeans .
<code>grid</code>	the reference grid supplied by emmeans .
<code>vcov.method</code>	character string specifying the covariance matrix estimator, one of the estimators described in vcov.geer , or a covariance matrix or function accepted by emmeans . Defaults to "bias-corrected".
<code>cov_type</code>	optional alias of <code>vcov.method</code> , which it overrides. Defaults to NULL.
<code>vcov.</code>	optional covariance matrix or function supplied through the emmeans argument <code>vcov.</code> , which overrides <code>cov_type</code> and <code>vcov.method</code> . Defaults to NULL.
<code>misc</code>	optional list passed through by emmeans . Defaults to NULL.

Value

`recover_data.geer()` returns the predictor data, and `emm_basis.geer()` the components from which **emmeans** builds a reference grid.

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,
    repeated = period,
    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 identifier of the 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. Carey and Wang (2011) use these data to compare working covariance structures.

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

<code>object</code>	a fitted model object of class "geer".
<code>...</code>	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Value

A numeric vector of fitted means, one per observation used in the fit, in the order given under Observation order.

Observation order

The values follow the internal order of the fitted model: the complete cases, sorted by cluster and, within cluster, by repeated. Unless the data were already in this order and had no missing values, assigning the values to the data, as in `data$r <- residuals(fit)`, misaligns the rows without warning. `fit$id` and `fit$repeated` are stored in the same order and can be used to align them.

See Also

[residuals.geer](#), [predict.geer](#).

Examples

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

```
frechet_bounds_cor
```

Fréchet Bounds for a Working Correlation Matrix

Description

Checks whether each working correlation of a fitted binary GEE model lies within the Fréchet bounds implied by the fitted marginal probabilities.

Usage

```
frechet_bounds_cor(object)
```

Arguments

object a fitted model object of class "geer", obtained from [geewa](#) with a binomial or quasibinomial family, a binary (0/1) response, and a working correlation structure other than "independence".

Details

For observations at times j and k of cluster i , with fitted probabilities π_{ij} and π_{ik} , the Fréchet 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 of a time pair is common to all clusters, whereas the bounds vary with the fitted probabilities. The bounds are undefined for fitted probabilities of 0 or 1, which usually indicate separation, and such fits give an error. Grouped binomial responses are not allowed. Clusters of size one are skipped, and time pairs observed in no cluster are omitted.

Value

A data frame with one row per time pair and columns:

alpha_name	the label alpha_j.k of the working correlation.
alpha_value	the working correlation.
lower_max	the largest lower bound over the clusters.
upper_min	the smallest upper bound over the clusters.
n_clusters	the number of clusters observed at both times.
n_violated	the number of clusters whose bounds do not contain alpha_value.

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"
)
frechet_bounds_cor(fit)
```

geecriteria

*Model Selection Criteria for geer Objects***Description**

Computes model selection criteria for one or more fitted `geer` objects, for comparing marginal mean models and for selecting the working association structure.

Usage

```
geecriteria(object, ..., criteria = "all", cov_type = "robust", digits = 2)
```

Arguments

<code>object</code>	a fitted model object of class <code>"geer"</code> .
<code>...</code>	additional fitted model objects of class <code>"geer"</code> to be included in the comparison.
<code>criteria</code>	character vector of criterion names (see Details), matched ignoring case, or <code>"all"</code> . Only the requested criteria are computed and they are returned in the order requested. Defaults to <code>"all"</code> .
<code>cov_type</code>	character string specifying the covariance matrix estimator used in the covariance-based penalty terms: one of the estimators described in vcov.geer . Defaults to <code>"robust"</code> , which gives the classical definitions of the criteria.
<code>digits</code>	nonnegative integer giving the number of decimal places used to round the criteria. Defaults to 2.

Details

The available criteria are:

QIC, QICu quasi-likelihood information criteria of Pan (2001).

QICHH QIC as modified by Hardin and Hilbe (2003, 2013).

QICC corrected QIC of Hardin and Hilbe (2013).

CIC correlation information criterion of Hin and Wang (2009).

RJC Rotnitzky-Jewell criterion of Hin, Carey and Wang (2007), based on Rotnitzky and Jewell (1990).

EQIC extended quasi-likelihood information criterion of Wang and Hin (2010).

GEISS generalized error sum of squares criterion of Shults and Chaganty (1998) and Shults et al. (2009).

GPC, AGPC, SGPC Gaussian pseudolikelihood criterion and its Akaike-type and Schwarz-type penalized versions (Carey and Wang, 2011; Zhu and Zhu, 2013; Fu, Hao and Wang, 2018).

GHYC criterion of Gosho, Hamada and Yoshimura (2011), denoted DEW by Gosho (2014).

PAC criterion of Pardo and Alonso (2019).

PT, WR, RMR Pillai trace, Wilks ratio, and Roy maximum root type criteria of Jang (2011), also described by Pardo and Alonso (2019, Sections 3.6 to 3.8).

Smaller values are preferred for every criterion except GPC, for which larger values are preferred. QIC and QICu are intended mainly for comparing marginal mean models, and CIC, RJC, GHYC, and PAC for selecting the working association structure. The column Parameters, the number of regression parameters, is always reported.

cov_type affects QIC, QICHH, QICC, CIC, RJC, EQIC, PT, WR, and RMR; with cov_type = "jackknife" each model is refitted once per cluster. A criterion that cannot be evaluated for a fit is NA. A warning is issued when the models differ in the number of observations, the family, or the response values, because the criteria are then not comparable.

Let K be the number of clusters, n^* the number of observations, p the number of regression parameters, m the number of association parameters estimated from the data ($m = 0$ for independence and for corstr = "fixed" or orstr = "fixed"), $\hat{V}$ the covariance estimate selected by cov_type and $\hat{\Omega}_I$ the model-based information matrix under working independence. Then:

- $QIC_{HH} = -2\Psi(\hat{\beta}_I; I) + 2\text{tr}(\hat{\Omega}_I \hat{V})$, where $\Psi(\hat{\beta}_I; I)$ is the quasi-likelihood at $\hat{\beta}_I$, the ordinary GEE estimate under working independence, refitted from the stored data.
- $QICC = QIC - 2(p+m)(p+m+1)/(K-p-m-1)$, and NA when $K-p-m-1 \leq 0$. This follows equation (4.30) of Hardin and Hilbe (2013); the **geepack** package adds the correction instead.
- GEISS is the error sum of squares weighted by the inverse working covariance matrices, divided by $n^* - p - m$. Without this divisor it is the criterion $SC(R)$ of Pardo and Alonso (2019), and the two can rank structures differently.
- With $D_G = n^* \log(2\pi) - 2$ GPC, AGPC = $D_G + 2(p+m)$ and SGPC = $D_G + \log(K)(p+m)$, on the scale used by **glmtoolbox** (Vanegas, Rondon and Paula, 2023).
- With λ_j the eigenvalues of $\hat{V}\hat{\Omega}_I$, $PT = \sum_j \lambda_j/(1 + \lambda_j)$, $WR = \prod_j \lambda_j/(1 + \lambda_j)$, and $RMR = \max_j \lambda_j/(1 + \lambda_j)$. They are NA if an eigenvalue is not positive.

- With $\bar{S} = K^{-1} \sum_i (y_i - \hat{\mu}_i)(y_i - \hat{\mu}_i)^\top$ and $\bar{V} = K^{-1} \sum_i \hat{V}_i$, $\text{GHYC} = \text{tr}[(\bar{S}\bar{V}^{-1} - I)^2]$ and $\text{PAC} = |\det(\bar{S})/\det(\bar{V}) - 1|$. Both require a balanced design, in which every cluster observes each repeated position once, and are NA otherwise or when $\bar{V}$ is singular.
- EQIC uses the extended quasi-likelihood with the Nelder-Pregibon adjustment $k = 1/6$. Its dispersion is 1 for binomial models, the fixed value when `phi_fixed = TRUE`, and otherwise the adjusted deviance divided by n^* .

The quasi-likelihood in QIC, QICu, QICHH, and QICC diverges when a fitted mean reaches the boundary of its support, for example a fitted probability of 0 or 1. Fitted means are therefore kept `sqrt(.Machine$double.eps)` away from the boundary. This does not affect the comparison of association structures for the same mean model, but these criteria should not be used to compare mean models when only some of them reach the boundary; use CIC instead.

Value

A data frame with one row per model, containing the requested criteria and `Parameters`. When more than one model is supplied, the row names are the deparsed model expressions, or "Model 1", "Model 2", and so on for models supplied as objects rather than expressions, for example through `do.call`.

References

- Carey, V.J. and Wang, Y.G. (2011) Working covariance model selection for generalized estimating equations. *Statistics in Medicine*, **30**, 3117–3124.
- Fu, L., Hao, Y. and Wang, Y.G. (2018) Working correlation structure selection in generalized estimating equations. *Computational Statistics*, **33**, 983–996.
- Gosho, M. (2014) Criteria to select a working correlation structure for the generalized estimating equations method in SAS. *Journal of Statistical Software, Code Snippets*, **57**, 1–10.
- Gosho, M., Hamada, C. and Yoshimura, I. (2011) Criterion for the selection of a working correlation structure in the generalized estimating equation approach for longitudinal balanced data. *Communications in Statistics – Theory and Methods*, **40**, 3839–3856.
- Hardin, J.W. and Hilbe, J.M. (2003) *Generalized Estimating Equations*. Chapman and Hall/CRC, Boca Raton.
- Hardin, J.W. and Hilbe, J.M. (2013) *Generalized Estimating Equations*, 2nd Edition. Chapman and Hall/CRC, Boca Raton.
- Hin, L.Y., Carey, V.J. and Wang, Y.G. (2007) Criteria for working-correlation-structure selection in GEE: assessment via simulation. *The American Statistician*, **61**, 360–364.
- Hin, L.Y. and Wang, Y.G. (2009) Working-correlation-structure identification in generalized estimating equations. *Statistics in Medicine*, **28**, 642–658.
- Jang, M.J. (2011) *Working correlation selection in generalized estimating equations*. PhD dissertation, University of Iowa.
- Pan, W. (2001) Akaike's information criterion in generalized estimating equations. *Biometrics*, **57**, 120–125.
- Pardo, M.C. and Alonso, R. (2019) Working correlation structure selection in GEE analysis. *Statistical Papers*, **60**, 1447–1467.

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.

Shults, J. and Chaganty, N.R. (1998) Analysis of serially correlated data using quasi-least squares. *Biometrics*, **54**, 1622–1630.

Shults, J., Sun, W., Tu, X., Kim, H., Amsterdam, J., Hilbe, J.M. and Ten-Have, T. (2009) A comparison of several approaches for choosing between working correlation structures in generalized estimating equation analysis of longitudinal binary data. *Statistics in Medicine*, **28**, 2338–2355.

Vanegas, L.H., Rondon, L.M. and Paula, G.A. (2023) Generalized Estimating Equations using the new R package glmttoolbox. *The R Journal*, **15**, 105–133.

Wang, Y.G. and Hin, L.Y. (2010) Modeling strategies in longitudinal data analysis: covariate, variance function and correlation structure selection. *Computational Statistics & Data Analysis*, **54**, 3359–3370.

Zhu, X. and Zhu, Z. (2013) Comparison of criteria to select working correlation matrix in generalized estimating equations. *Chinese Journal of Applied Probability and Statistics*, **29**, 515–530.

See Also

[vcov.geer](#), [step_criterion](#), [step_p](#), [glance.geer](#), [anova.geer](#), [geewa](#), [geewa_binary](#).

Examples

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

## Selected criteria, in the order requested
geecriteria(fit_ind, fit, fit_ar1, criteria = c("cic", "QIC"))

## Compare estimation methods
fit_br <- update(fit, method = "brgee-robust")
geecriteria(fit, fit_br)
```

geer_control	<i>Control Parameters for geer Model Fitting</i>
--------------	--

Description

Sets the control parameters of the fitting algorithm of [geewa](#) and [geewa_binary](#).

Usage

```
geer_control(
  tolerance = 1e-06,
  maxiter = 500L,
  or_adding = 0.5,
  step_maxiter = 10L,
  step_multiplier = 1,
  jeffreys_power = 0.5
)
```

Arguments

tolerance	strictly positive number giving the convergence tolerance. Defaults to 1e-06.
maxiter	positive integer giving the maximum number of iterations. Defaults to 500.
or_adding	strictly positive number added to each cell of the marginalized 2×2 tables from which geewa_binary estimates the working odds ratios. Defaults to 0.5.
step_maxiter	positive integer giving the maximum number of step-halvings in an iteration. Defaults to 10.
step_multiplier	strictly positive number scaling the proposed step before step-halving. Values below 1 damp the step, which can help when the algorithm diverges from poor starting values. Defaults to 1.
jeffreys_power	strictly positive number giving the power of the Jeffreys-type penalty of the methods "pgee-jeffreys", "opgee-jeffreys", and "hpgee-jeffreys" (Touloumis, 2026b). Defaults to 0.5, the standard Jeffreys-type penalty.

Value

A list with components tolerance, maxiter, or_adding, step_maxiter, step_multiplier, and jeffreys_power.

References

Touloumis, A. (2026b) Jeffreys-type penalized GEE for correlated binary data with an odds-ratio parameterization. *arXiv preprint*. doi:[10.48550/arXiv.2606.16058](https://doi.org/10.48550/arXiv.2606.16058)

See Also

[geewa](#), [geewa_binary](#).

Examples

```
## Default control parameters
geer_control()

## Tighter tolerance and fewer iterations
geer_control(tolerance = 1e-08, maxiter = 200)

## Larger continuity correction for the odds-ratio estimates
geer_control(or_adding = 1)

## Weaker Jeffreys-type penalty
geer_control(jeffreys_power = 0.1)

## Damped steps, useful when the algorithm diverges
geer_control(step_multiplier = 0.5)
```

geewa

Fitting (Adjusted) Generalized Estimating Equations

Description

Fits a marginal model for clustered or repeated responses by generalized estimating equations (GEE), using ordinary, bias-reduced, bias-corrected, or Jeffreys-type penalized GEE.

Usage

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

Arguments

formula	formula of the form <code>response ~ predictors</code> describing the marginal mean model. Terms <code>offset()</code> are allowed.
family	family object, family function, or character string naming a family, specifying the marginal variance and link functions. Supported families are <code>gaussian</code> , <code>binomial</code> , <code>poisson</code> , <code>Gamma</code> , <code>inverse.gaussian</code> , <code>quasi</code> , <code>quasibinomial</code> , and <code>quasipoisson</code> . Defaults to <code>gaussian(link = "identity")</code> .
data	optional data frame in long format, with one row per observation, containing the variables in formula, <code>id</code> , <code>repeated</code> , <code>weights</code> , and <code>offset</code> . Defaults to <code>parent.frame()</code> .
id	variable identifying the clusters.
repeated	optional variable identifying the order of the observations within each cluster. Its values must be unique within each cluster. They are replaced by their rank among the sorted distinct values, or by the level number for a factor, so only their order matters and not their spacing. A character variable is sorted alphabetically, so <code>"10"</code> precedes <code>"2"</code> , and gives a warning. If omitted, the row order within each cluster is used.
control	list of control parameters as returned by geer_control . Defaults to <code>geer_control(...)</code> .
corstr	character string specifying the working correlation structure: <code>"independence"</code> , <code>"exchangeable"</code> , <code>"ar1"</code> , <code>"m-dependent"</code> , <code>"unstructured"</code> , <code>"toeplitz"</code> , or <code>"fixed"</code> . Defaults to <code>"independence"</code> .
Mv	positive integer giving the order of the <code>"m-dependent"</code> structure. Ignored for other structures. Defaults to 1.
method	character string specifying the estimation method: ordinary GEE (<code>"gee"</code>), bias-reduced GEE (<code>"brgee-naive"</code> , <code>"brgee-robust"</code> , or <code>"brgee-empirical"</code>), bias-corrected GEE (<code>"bcgee-naive"</code> , <code>"bcgee-robust"</code> , or <code>"bcgee-empirical"</code>), Jeffreys-type penalized GEE (<code>"pgee-jeffreys"</code>), or its one-step (<code>"opgee-jeffreys"</code>) and hybrid one-step (<code>"hpgee-jeffreys"</code>) versions. See Details. Defaults to <code>"gee"</code> .
weights	optional numeric vector of strictly positive, finite observation weights. Defaults to weights of 1.
beta_start	optional numeric vector of starting values for the regression parameters. Defaults to <code>NULL</code> , in which case they are obtained from a generalized linear model fitted by brglmFit .
offset	optional a priori known component of the linear predictor: a single number or a numeric vector with one value per observation. It is added to any <code>offset()</code> terms in formula.
control_glm	optional list of control parameters passed to brglm_control for the starting values. Ignored when <code>beta_start</code> is supplied. Defaults to <code>list(...)</code> .
use_p	logical indicating whether the number of regression parameters is subtracted from the denominators of the estimators of the dispersion and working correlation parameters. Defaults to <code>TRUE</code> .

<code>alpha_vector</code>	numeric vector of working correlation parameters for <code>constr = "fixed"</code> , of length <code>choose(T, 2)</code> , where T is the number of distinct repeated values, giving a positive definite correlation matrix. Ignored for other structures. Defaults to <code>NULL</code> .
<code>phi_fixed</code>	logical indicating whether the dispersion parameter is fixed at <code>phi_value</code> . Defaults to <code>FALSE</code> .
<code>phi_value</code>	strictly positive number giving the fixed value of the dispersion parameter when <code>phi_fixed = TRUE</code> . Defaults to 1.
<code>subset</code>	optional vector specifying a subset of observations to be used in the fit, evaluated in data as in <code>model.frame</code> .
<code>na.action</code>	function specifying the handling of missing values, as in <code>model.frame</code> . Defaults to <code>getOption("na.action")</code> .
<code>...</code>	arguments used to build <code>control</code> and <code>control_glm</code> when these are not supplied. Supplying <code>...</code> together with both <code>control</code> and <code>control_glm</code> is an error.

Details

The estimation methods are:

"gee" the estimating equations of Liang and Zeger (1986).

"brgee-*" bias-reduced GEE, which add an adjustment term to the estimating equations based on the naive, robust, or empirical covariance matrix (Touloumis, 2026a).

"bcgee-*" bias-corrected GEE, which apply a one-step correction to the converged GEE estimate (Touloumis, 2026a).

"pgee-jeffreys" GEE penalized by a Jeffreys-type penalty (Touloumis, 2026b).

"opgee-jeffreys", "hpgee-jeffreys" one penalized, or one ordinary, scoring step from the penalized estimate under working independence.

Observations are sorted by `id` and `repeated`, and both are recoded to consecutive integers. The quasi families are mapped to the corresponding standard families: `quasibinomial` to `binomial`, `quasipoisson` to `poisson`, and quasi with variance `"constant"`, `"mu(1-mu)"`, `"mu"`, `"mu^2"`, or `"mu^3"` to `gaussian`, `binomial`, `poisson`, `Gamma`, or `inverse.gaussian`. The dispersion parameter is estimated for every family unless `phi_fixed = TRUE`.

For `constr = "ar1"`, the correlation parameter is estimated from the products of Pearson residuals at consecutive repeated values, pooled over clusters, so it can differ from the estimates of other packages.

Value

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

<code>coefficients</code>	a named vector of estimated regression coefficients.
<code>residuals</code>	the working residuals.
<code>fitted.values</code>	the fitted mean values, obtained by transforming the linear predictors by the inverse of the link function.
<code>rank</code>	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.
cluster_labels	the cluster identifiers supplied, in the order of their recoded values in id.
repeated_labels	the values of repeated supplied, in the order of their recoded values in repeated, or NULL if repeated was not supplied.
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.
fit_function	character string identifying the fitting function used, either "geewa" or "geewa_binary".
use_p	logical indicating whether the degrees-of-freedom correction was used; geewa fits only.
phi_fixed	logical indicating whether the dispersion parameter was fixed; geewa fits only.
contrasts	the contrasts used.
xlevels	a record of the levels of the factors used in fitting.
na.action	the na.action attribute of the model frame, present only when 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	the estimated or fixed working association parameters, 0 under independence.

`phi` the estimated or fixed dispersion parameter, 1 for `geewa_binary` fits.
`obs_no` the number of observations used in fitting.
`clusters_no` the number of clusters.
`min_cluster_size` the minimum cluster size.
`max_cluster_size` the maximum cluster size.

Other components are for internal use.

Convergence and separation

For the two-stage methods ("`bcgee-*`", "`opgee-jeffreys`", and "`hpgee-jeffreys`"), `alpha` and `phi` are estimated once at the first-stage estimate and held fixed in the correction step, and `converged` is `TRUE` unless the correction step fails, in which case the first-stage estimates are returned with a warning.

If `converged` is `FALSE`, the results are those of the last accepted iteration and should not be interpreted; `print()` and `summary()` then show a warning. `converged = TRUE` only means that the stopping criterion was met: with separated binary data or all-zero counts the estimates can be very large, which is signaled by a warning that fitted probabilities are within $1e-12$ of 0 or 1 or that fitted rates are 0.

In the bias-reduced and penalized methods the adjustment term is multiplied by the dispersion parameter, whose estimate tends to 0 under complete separation, so with the default estimated dispersion no method protects against infinite estimates. For binary responses with separation or rare events, use `phi_fixed = TRUE` or `geewa_binary`. Then "`brgee-naive`" and the Jeffreys-penalized methods ("`pgee-jeffreys`", "`opgee-jeffreys`", and "`hpgee-jeffreys`") keep the estimates finite under separation. "`brgee-robust`" and "`brgee-empirical`" do not, and the bias-corrected methods fail because they start from the ordinary GEE fit.

References

- Liang, K.Y. and Zeger, S.L. (1986) Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 13–22.
- Touloumis, A. (2026a) Bias-reduced GEE via adjusted estimating equations, with odds-ratio extensions. *arXiv preprint*. doi:10.48550/arXiv.2606.16043
- Touloumis, A. (2026b) Jeffreys-type penalized GEE for correlated binary data with an odds-ratio parameterization. *arXiv preprint*. doi:10.48550/arXiv.2606.16058

See Also

`geewa_binary`, `geer_control`, `summary.geer`, `vcov.geer`, `anova.geer`, `step_p`, `geecriteria`, `runs_test`.

Examples

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

```

    family = poisson(link = "log"),
    data = epilepsy,
    id = id,
    repeated = visit,
    corstr = "exchangeable",
    method = "gee"
)
summary(fit)

## Bias-reduced and bias-corrected GEE
fit_br <- update(fit, method = "brgee-robust")
summary(fit_br)
fit_bc <- update(fit, method = "bcgee-robust")
summary(fit_bc, cov_type = "robust")

## Jeffreys-type penalized GEE with a weaker penalty
fit_pgee <- update(fit, method = "pgee-jeffreys",
                  control = geer_control(jeffreys_power = 0.1))
summary(fit_pgee)

```

geewa_binary

Fitting (Adjusted) Generalized Estimating Equations for Binary Responses

Description

Fits a marginal model for clustered or repeated binary responses by generalized estimating equations (GEE), with the association between responses modeled by marginalized pairwise odds ratios, using ordinary, bias-reduced, bias-corrected, or Jeffreys-type 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,

```

```

subset,
na.action,
...
)

```

Arguments

formula	formula of the form <code>response ~ predictors</code> describing the marginal mean model. Terms <code>offset()</code> are allowed.
link	character string specifying the link function: "logit", "probit", "cauchit", "cloglog", "identity", "log", "sqrt", "1/mu^2", or "inverse". Defaults to "logit".
data	optional data frame in long format, with one row per observation, containing the variables in formula, id, repeated, weights, and offset. Defaults to <code>parent.frame()</code> .
id	variable identifying the clusters.
repeated	optional variable identifying the order of the observations within each cluster. Its values must be unique within each cluster. They are replaced by their rank among the sorted distinct values, or by the level number for a factor, so only their order matters and not their spacing. A character variable is sorted alphabetically, so "10" precedes "2", and gives a warning. If omitted, the row order within each cluster is used.
control	list of control parameters as returned by <code>geer_control</code> . Defaults to <code>geer_control(...)</code> .
orstr	character string specifying the working odds-ratio structure: "independence", "exchangeable", "unstructured", or "fixed". Defaults to "independence".
method	character string specifying the estimation method: ordinary GEE ("gee"), bias-reduced GEE ("brgee-naive", "brgee-robust", or "brgee-empirical"), bias-corrected GEE ("bcgee-naive", "bcgee-robust", or "bcgee-empirical"), Jeffreys-type penalized GEE ("pgee-jeffreys"), or its one-step ("opgee-jeffreys") and hybrid one-step ("hpgee-jeffreys") versions. See Details. Defaults to "gee".
weights	optional numeric vector of strictly positive, finite observation weights. Defaults to weights of 1.
beta_start	optional numeric vector of starting values for the regression parameters. Defaults to NULL, in which case they are obtained from a generalized linear model fitted by <code>brglmFit</code> .
offset	optional a priori known component of the linear predictor: a single number or a numeric vector with one value per observation. It is added to any <code>offset()</code> terms in formula.
control_glm	optional list of control parameters passed to <code>brglm_control</code> for the starting values. Ignored when <code>beta_start</code> is supplied. Defaults to <code>list(...)</code> .
alpha_vector	numeric vector of strictly positive, finite odds ratios for <code>orstr = "fixed"</code> , of length <code>choose(T, 2)</code> , where T is the number of distinct repeated values. Ignored for other structures. Defaults to NULL.

subset	optional vector specifying a subset of observations to be used in the fit, evaluated in data as in <code>model.frame</code> .
na.action	function specifying the handling of missing values, as in <code>model.frame</code> . Defaults to <code>getOption("na.action")</code> .
...	arguments used to build <code>control</code> and <code>control_glm</code> when these are not supplied. Supplying ... together with both <code>control</code> and <code>control_glm</code> is an error.

Details

The estimation methods are described in [geewa](#). The marginal mean model uses the binomial family with the given link, the working odds ratios are estimated from marginalized 2×2 tables (see `or_adding` in [geer_control](#)), and the dispersion parameter is fixed at 1. Observations are sorted by `id` and `repeated`, and both are recoded to consecutive integers.

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.
cluster_labels	the cluster identifiers supplied, in the order of their recoded values in <code>id</code> .
repeated_labels	the values of <code>repeated</code> supplied, in the order of their recoded values in <code>repeated</code> , or NULL if <code>repeated</code> was not supplied.
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 <code>geer_control</code> list used.
method	character string identifying the estimation method used.
fit_function	character string identifying the fitting function used, either "geewa" or "geewa_binary".
use_p	logical indicating whether the degrees-of-freedom correction was used; <code>geewa</code> fits only.
phi_fixed	logical indicating whether the dispersion parameter was fixed; <code>geewa</code> fits only.
contrasts	the contrasts used.
xlevels	a record of the levels of the factors used in fitting.
na.action	the <code>na.action</code> attribute of the model frame, present only when 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	the estimated or fixed working association parameters, 0 under independence.
phi	the estimated or fixed dispersion parameter, 1 for <code>geewa_binary</code> fits.
obs_no	the number of observations used in fitting.
clusters_no	the number of clusters.
min_cluster_size	the minimum cluster size.
max_cluster_size	the maximum cluster size.

Other components are for internal use.

Convergence and separation

Here `phi` is always 1 and `alpha` contains the working odds ratios, which are 1 under independence and are computed once from the data, so they are the same in both stages of the two-stage methods.

For the two-stage methods ("`bcgee-*`", "`opgee-jeffreys`", and "`hpgee-jeffreys`"), `converged` is TRUE unless the correction step fails, in which case the first-stage estimates are returned with a warning. If `converged` is FALSE, the results are those of the last accepted iteration and should not be interpreted. `converged = TRUE` only means that the stopping criterion was met: with separated data the estimates can be very large, which is signaled by a warning that fitted probabilities are within $1e-12$ of 0 or 1. Because the dispersion is fixed, "`brgee-naive`" and the Jeffreys-penalized methods ("`pgee-jeffreys`", "`opgee-jeffreys`", and "`hpgee-jeffreys`") keep the estimates finite under separation. "`brgee-robust`" and "`brgee-empirical`" do not, and the bias-corrected methods fail because they start from the ordinary GEE fit.

References

- Touloumis, A. (2026a) Bias-reduced GEE via adjusted estimating equations, with odds-ratio extensions. *arXiv preprint*. doi:10.48550/arXiv.2606.16043
- Touloumis, A. (2026b) Jeffreys-type penalized GEE for correlated binary data with an odds-ratio parameterization. *arXiv preprint*. doi:10.48550/arXiv.2606.16058

See Also

[geewa](#), [geer_control](#), [summary.geer](#), [vcov.geer](#), [anova.geer](#), [step_p](#), [geecriteria](#), [runs_test](#).

Examples

```
data("cholecystectomy", package = "geer")
fit <- geewa_binary(
  formula = pain ~ treatment + gender + age,
  link = "logit",
  data = cholecystectomy,
  id = id,
  repeated = time,
  orstr = "independence",
  method = "gee"
)
summary(fit)

## Bias-reduced and bias-corrected GEE
fit_br <- update(fit, method = "brgee-robust")
summary(fit_br)
fit_bc <- update(fit, method = "bcgee-robust")
summary(fit_bc, cov_type = "robust")

## Jeffreys-type penalized GEE with an exchangeable odds-ratio structure
fit_pgee <- update(fit, orstr = "exchangeable", method = "pgee-jeffreys")
summary(fit_pgee)

## Probit link
fit_probit <- update(fit, link = "probit")
summary(fit_probit)
```

Description

Summarizes a fitted geer object in a one-row data frame, following the conventions of **broom** (Robinson, Hayes and Couch, 2024).

Usage

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

Arguments

`x` a fitted model object of class "geer".

`...` not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Details

QIC and CIC use the robust covariance matrix, as in `geecriteria(x, cov_type = "robust")`, and are NA if they cannot be computed. See [geecriteria](#) for further criteria.

Value

A one-row data frame, a tibble if **tibble** is installed, with columns:

<code>family, link</code>	the family and link function.
<code>method</code>	the estimation method.
<code>wastr</code>	the working association structure, <code>corstr</code> or <code>orstr</code> .
<code>nobs, nclusters</code>	the numbers of observations and clusters.
<code>min.cluster.size, max.cluster.size</code>	the smallest and largest cluster sizes.
<code>npair</code>	the number of regression parameters.
<code>df.residual</code>	the number of observations minus <code>npair</code> .
<code>phi</code>	the dispersion parameter, 1 for geewa_binary fits.
<code>QIC, QICu</code>	the quasi-likelihood information criteria of Pan (2001).
<code>CIC</code>	the correlation information criterion of Hin and Wang (2009).
<code>converged</code>	logical indicating whether the algorithm converged.
<code>niter</code>	the number of iterations.

References

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.

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

See Also

[tidy.geer](#), [geecriteria](#), [geewa](#), [geewa_binary](#).

Examples

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

## Compare working correlation structures
fit_ar1 <- update(fit, corstr = "ar1")
rbind(glance(fit), glance(fit_ar1))
```

gof_residual_test	<i>Residual Goodness-of-Fit Tests for Binary geer Fits</i>
-------------------	--

Description

Tests the fit of the marginal mean model of a binary response with the Pearson chi-squared statistic or the unweighted sum of squared residuals, standardized by their approximate means and variances (Pan, 2002).

Usage

```
gof_residual_test(
  object,
  statistic = c("pearson", "uss"),
  covariance = c("unstructured", "empirical")
)
```

Arguments

object	a fitted model object of class "geer".
statistic	character string specifying the statistic: the Pearson chi-squared statistic ("pearson") or the unweighted sum of squared residuals ("uss"). Defaults to "pearson".
covariance	character string specifying the estimator of the covariance matrix of the responses used in the variance of the statistic: the pooled unstructured estimator ("unstructured") or the empirical estimator ("empirical"). See Details. Defaults to "unstructured".

Details

With fitted probabilities $\hat{\mu}_{ij}$, the statistics are

$$G = \sum_{i,j} \frac{(y_{ij} - \hat{\mu}_{ij})^2}{\hat{\mu}_{ij}(1 - \hat{\mu}_{ij})}, \quad U = \sum_{i,j} (y_{ij} - \hat{\mu}_{ij})^2,$$

with approximate means N , the number of observations, and $\sum_{i,j} \hat{\mu}_{ij}(1 - \hat{\mu}_{ij})$. Their approximate variances are $c^\top (I - H) \text{cov}(Y) (I - H)^\top c$, where c has elements $(1 - 2\hat{\mu}_{ij})/(\hat{\mu}_{ij}(1 - \hat{\mu}_{ij}))$ for G and $1 - 2\hat{\mu}_{ij}$ for U , and $H = D(\sum_i D_i^\top V_i^{-1} D_i)^{-1} D^\top V^{-1}$ is the leverage matrix of [hatvalues.geer](#). The standardized statistic is compared with the standard normal distribution, and the p-value is two-sided, which reproduces the p-values reported by Pan (2002) and Hosmer et al. (1997).

With covariance = "unstructured", $\text{cov}(Y_i)$ is estimated by $A_i^{1/2} \hat{R} A_i^{1/2}$, where A_i is the diagonal matrix of the $\hat{\mu}_{ij}(1 - \hat{\mu}_{ij})$ and $\hat{R}$ the pooled correlation matrix of the standardized residuals. Each element of $\hat{R}$ is the average of the products of the standardized residuals at the two repeated positions over the clusters that observe both. Unlike the pooled estimators of [vcov.geer](#), such as `cov_type = "pan"`, which require a balanced design, $\hat{R}$ is therefore available for unbalanced designs. Pan (2002) studied balanced designs, and with unbalanced designs the positions should be given by repeated. With "empirical", the covariance is estimated by $(y_i - \hat{\mu}_i)(y_i - \hat{\mu}_i)^\top$. Pan (2002) recommends the unstructured estimator. Pan (2002) and Evans and Li (2005) label the two versions of each statistic in opposite ways.

Pan (2002) derives the variances for the logit link and recommends a working independence fit, under which the approximation of the residuals is most accurate. Here the working covariance matrices V_i of object are used, and $D_i = \partial \mu_i / \partial \beta^\top$ for any link, so object can be a [geewa](#) fit of a binomial family or a [geewa_binary](#) fit. The response must be binary with unit prior weights. The tests are intended for data with many covariate patterns, such as when the model contains a continuous covariate. With few covariate patterns, the variance of the statistic is close to zero, the Pearson and unweighted sum of squares statistics can give the same p-value, and the tests are not informative.

The tests were developed for method = "gee". For the other estimation methods, the statistic is computed from the fitted values of object and a warning is issued, since its approximate mean and variance have not been studied for these estimators.

Value

An object of class `c("geer_hstest", "hstest")`. The standard components hold the standardized statistic (`statistic`), the statistic (`estimate`), and its approximate mean (`null.value`). In addition, the object contains:

<code>mean</code>	the approximate mean of the statistic.
<code>variance</code>	the approximate variance of the statistic.
<code>covariance</code>	the estimator of the covariance matrix of the responses used.

References

Evans, S. and Li, L. (2005) A comparison of goodness of fit tests for the logistic GEE model. *Statistics in Medicine*, **24**, 1245–1261.

Hosmer, D.W., Hosmer, T., le Cessie, S. and Lemeshow, S. (1997) A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in Medicine*, **16**, 965–980.

Pan, W. (2002) Goodness-of-fit tests for GEE with correlated binary data. *Scandinavian Journal of Statistics*, **29**, 101–110.

See Also

[gof_score_test](#), [gof_tests](#), [hatvalues.geer](#), [geewa](#), [geewa_binary](#).

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", ]
fit <- geewa_binary(
  formula = status ~ baseline + treatment + gender + visit + age,
  link = "logit",
  data = respiratory2,
  id = id,
  repeated = visit,
  orstr = "independence"
)
gof_residual_test(fit)
gof_residual_test(fit, statistic = "uss", covariance = "empirical")
```

gof_score_test

Partition Goodness-of-Fit Tests for Binary geer Fits

Description

Tests the fit of the marginal mean model of a binary response by adding indicators of a partition of the observations to the linear predictor and testing them with a score statistic: the deciles-of-risk test of Horton et al. (1999) or the covariate-region test of Barnhart and Williamson (1998).

Usage

```
gof_score_test(
  object,
  partition = c("risk", "covariates"),
  n_groups = 10L,
  regions,
  by_repeated = TRUE,
  cov_type = "bias-corrected"
)
```

Arguments

<code>object</code>	a fitted model object of class "geer".
<code>partition</code>	character string specifying the partition: groups of the fitted probabilities ("risk") or regions of the covariates ("covariates"). See Details. Defaults to "risk".
<code>n_groups</code>	a single integer, at least 2, giving the number of groups of fitted probabilities for <code>partition = "risk"</code> . Defaults to 10.
<code>regions</code>	for <code>partition = "covariates"</code> , a one-sided formula whose variables, evaluated in the data used to fit the model and then in the environment of the formula, define the regions by their combinations, or a vector with one value per observation, in the order stored in <code>object</code> . Continuous covariates should be cut into intervals, for example with <code>cut</code> .
<code>by_repeated</code>	logical indicating whether, for <code>partition = "covariates"</code> , the alternative model also contains indicators of the repeated positions and their interactions with the regions, as in Barnhart and Williamson (1998). Defaults to TRUE.
<code>cov_type</code>	character string specifying the covariance matrix estimator of the score statistic: one of the estimators described in <code>vcov.geer</code> , except "jackknife". See Details. Defaults to "bias-corrected".

Details

The alternative model adds to the linear predictor the indicators of the groups or regions, of which those linearly dependent on the model matrix are dropped. The test of their coefficients is the score test of `anova.geer` with `test = "score"`, evaluated at the fit of `object`, so the alternative model is not fitted. The association and dispersion parameters are those of `object`. The statistic is compared with the chi-squared distribution whose degrees of freedom are the number of added indicators kept.

With `partition = "risk"` (Horton et al., 1999), the fitted probabilities of all observations are divided at their empirical quantiles of orders $0, 1/G, \dots, 1$, where G is `n_groups`, into intervals closed on the right. Tied fitted probabilities fall into the same group, so the groups may have unequal sizes and, with few distinct fitted probabilities, there may be fewer than G groups. Implementations of Hosmer-Lemeshow type tests differ in this choice and may give different results (Hosmer et al., 1997). A warning is issued when fewer than G groups are formed. The groups are labeled by their intervals of fitted probabilities.

With `partition = "covariates"` (Barnhart and Williamson, 1998), the regions are the combinations of the variables of regions, so `~ treatment + gender` and `~ treatment:gender` define the same regions. With `by_repeated = TRUE`, the alternative model also contains the indicators of the repeated positions and of the region by position combinations, which is their model (4). The cut points should be chosen before looking at the data. Following Barnhart and Williamson (1998), a warning is issued when a region has fewer than 10 clusters, when more than 25% of the regions have fewer than 25 clusters, or when a region, or with `by_repeated = TRUE` a region by position cell, has no events or no non-events.

With `cov_type = "naive"`, the statistic is the model-based statistic Q of Barnhart and Williamson (1998), with the dispersion parameter of `object`. Barnhart and Williamson fix the dispersion parameter at 1, which a fit with `phi_fixed = TRUE` and `phi_value = 1` reproduces. With `cov_type = "robust"`, the statistic is the robust score statistic of Rotnitzky and Jewell (1990) used by Horton et al. (1999). This differs from the robust statistic Q_R of Barnhart and Williamson (1998), whose variance is the Schur complement of the robust middle matrix of the sandwich rather than

the variance of the score for the added coefficients at the fit of object. In their depression example, their Q_R of 38.93 on 8 degrees of freedom corresponds to 33.91 here. The default `cov_type = "bias-corrected"` is that of [anova.geer](#), and `cov_type = "robust"` gives the statistic of Horton et al. (1999).

The papers consider the logit link. The tests are computed with the derivative $D_i = \partial\mu_i/\partial\beta^\top$ for any link, and object can be a [geewa](#) fit of a binomial family or a [geewa_binary](#) fit. The response must be binary with unit prior weights.

The tests were developed for `method = "gee"`. For the other estimation methods, the score of the ordinary generalized estimating equations is evaluated at the fitted values of object, as in [anova.geer](#), and a warning is issued, since the null distribution of the statistic has not been studied for these estimators.

Value

An object of class `c("geer_htest", "htest")`. The standard components hold the score statistic (statistic) and its degrees of freedom (parameter). In addition, the object contains:

<code>partition</code>	a factor giving the group or region of each observation, in the order stored in object.
<code>table</code>	a data frame with one row per group or region and the numbers of observations and of clusters, and the observed and expected numbers of events.
<code>cov_type</code>	the covariance matrix estimator used.

References

- Barnhart, H.X. and Williamson, J.M. (1998) Goodness-of-fit tests for GEE modeling with binary responses. *Biometrics*, **54**, 720–729.
- Horton, N.J., Bebbchuk, J.D., Jones, C.L., Lipsitz, S.R., Catalano, P.J., Zahner, G.E.P. and Fitzmaurice, G.M. (1999) Goodness-of-fit for GEE: an example with mental health service utilization. *Statistics in Medicine*, **18**, 213–222.
- Hosmer, D.W., Hosmer, T., le Cessie, S. and Lemeshow, S. (1997) A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in Medicine*, **16**, 965–980.
- 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

[gof_residual_test](#), [gof_tests](#), [anova.geer](#), [geewa](#), [geewa_binary](#).

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", ]
fit <- geewa_binary(
  formula = status ~ baseline + treatment + gender + visit + age,
  link = "logit",
  data = respiratory2,
  id = id,
```

```

    repeated = visit,
    orstr = "exchangeable"
  )
  gof_score_test(fit)
  gof_score_test(fit, partition = "covariates", regions = ~treatment)
  gof_score_test(
    fit,
    partition = "covariates",
    regions = ~treatment,
    cov_type = "robust"
  )

```

gof_tests

Goodness-of-Fit Tests for Binary geer Fits

Description

Computes the goodness-of-fit tests of [gof_score_test](#) and [gof_residual_test](#) for a binary geer fit and collects them in one table.

Usage

```

gof_tests(
  object,
  n_groups = 10L,
  regions,
  by_repeated = TRUE,
  cov_type = "bias-corrected",
  covariance = c("unstructured", "empirical")
)

```

Arguments

object	a fitted model object of class "geer".
n_groups	a single integer, at least 2, giving the number of groups of fitted probabilities for the test of Horton et al. (1999). Defaults to 10.
regions	optional one-sided formula or vector defining covariate regions, as in gof_score_test . If supplied, the covariate-region tests of Barnhart and Williamson (1998) are included.
by_repeated	logical indicating whether the alternative model of the covariate-region tests also contains indicators of the repeated positions and their interactions with the regions, as in gof_score_test . Used only when regions is supplied. Defaults to TRUE.
cov_type	character string specifying the covariance matrix estimator of the score statistics, as in gof_score_test . Defaults to "bias-corrected".

covariance character string specifying the estimator of the covariance matrix of the responses used in the variance of the statistic: the pooled unstructured estimator ("unstructured") or the empirical estimator ("empirical"). See Details. Defaults to "unstructured".

Details

The table contains the deciles-of-risk test of Horton et al. (1999) with `cov_type`, the covariate-region tests of Barnhart and Williamson (1998) with the naive covariance matrix and with `cov_type` when regions is supplied, and the Pearson and unweighted sum of squares tests of Pan (2002) with `covariance`. The tests have power against different departures from the model (Evans and Li, 2005), so they complement each other, and a single non-significant test is weak evidence of fit.

Value

An object of class `c("geer_anova", "anova", "data.frame")` with one row per test and columns `Statistic`, `Df` (for the score tests), `Mean` and `Variance` (the approximate mean and variance of the residual statistics), and `Pr(Test)`, the p-value of the chi-squared test or of the two-sided normal test.

References

- Barnhart, H.X. and Williamson, J.M. (1998) Goodness-of-fit tests for GEE modeling with binary responses. *Biometrics*, **54**, 720–729.
- Evans, S. and Li, L. (2005) A comparison of goodness of fit tests for the logistic GEE model. *Statistics in Medicine*, **24**, 1245–1261.
- Horton, N.J., Bechuk, J.D., Jones, C.L., Lipsitz, S.R., Catalano, P.J., Zahner, G.E.P. and Fitzmaurice, G.M. (1999) Goodness-of-fit for GEE: an example with mental health service utilization. *Statistics in Medicine*, **18**, 213–222.
- Pan, W. (2002) Goodness-of-fit tests for GEE with correlated binary data. *Scandinavian Journal of Statistics*, **29**, 101–110.

See Also

[gof_score_test](#), [gof_residual_test](#), [geewa](#), [geewa_binary](#).

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", ]
fit <- geewa_binary(
  formula = status ~ baseline + treatment + gender + visit + age,
  link = "logit",
  data = respiratory2,
  id = id,
  repeated = visit,
  orstr = "independence"
)
gof_tests(fit)
gof_tests(fit, regions = ~treatment)
```

Description

Computes leverages, the change in the estimated regression coefficients when a cluster or an observation is deleted, and Cook's distances for a fitted geer object.

Usage

```
## S3 method for class 'geer'  
hatvalues(model, unit = c("clusters", "observations"), ...)
```

```
## S3 method for class 'geer'  
dfbeta(  
  model,  
  unit = c("clusters", "observations"),  
  method = c("one-step", "exact"),  
  ...  
)
```

```
## S3 method for class 'geer'  
dfbetas(  
  model,  
  unit = c("clusters", "observations"),  
  method = c("one-step", "exact"),  
  cov_type = "naive",  
  ...  
)
```

```
## S3 method for class 'geer'  
cooks.distance(  
  model,  
  unit = c("clusters", "observations"),  
  method = c("one-step", "exact"),  
  cov_type = "naive",  
  parm,  
  ...  
)
```

Arguments

model	a fitted model object of class "geer".
unit	character string specifying whether the diagnostics refer to clusters ("clusters") or to single observations ("observations"). Defaults to "clusters".

...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.
method	character string specifying how the deletion estimates are obtained: the one-step approximation of Preisser and Qaqish (1996) ("one-step") or a refit of the model without each cluster ("exact"). See Details. Defaults to "one-step".
cov_type	character string specifying the covariance matrix estimator used to standardize the deletion estimates: one of the estimators described in vcov.geer . Defaults to "naive".
parm	a specification of which coefficients enter Cook's distance, either a vector of numbers or a vector of names. If missing, all coefficients are used.

Details

Let D_i be the derivative of the mean vector of cluster i with respect to β , V_i its working covariance matrix, including the dispersion, and $\Sigma = (\sum_i D_i^\top V_i^{-1} D_i)^{-1}$ the naive covariance matrix. The leverage matrix of cluster i is

$$H_i = D_i \Sigma D_i^\top V_i^{-1}.$$

`hatvalues()` returns the diagonal elements of H_i for `unit = "observations"` and their sum, the trace of H_i , for `unit = "clusters"`. The leverages sum to the number of coefficients.

`dfbeta()` returns $\hat{\beta} - \hat{\beta}_{(i)}$, where $\hat{\beta}_{(i)}$ is the estimate without cluster or observation i . With `method = "one-step"`, it is a single Fisher scoring step from $\hat{\beta}$ with the association and dispersion parameters held at their full-data values (Preisser and Qaqish, 1996). For a cluster this is

$$\Sigma D_i^\top V_i^{-1} (I - H_i)^{-1} (y_i - \hat{\mu}_i),$$

as in equation (4.66) of Hardin and Hilbe (2013). For observation j of cluster i , the step uses the residual, the derivative and the variance of y_{ij} conditional on the remaining observations of the cluster under the working covariance matrix. This differs from equation (4.67) of Hardin and Hilbe (2013), which ignores the correlation with the rest of the cluster and agrees with it only under working independence. Both one-step approximations are exact for a Gaussian response with the identity link.

With `method = "exact"`, every cluster is deleted in turn and the model refitted with the association parameters held at their full-data values, as for `cov_type = "jackknife"` in [vcov.geer](#). The refits are computed once and cached on the fit. It is available only for `unit = "clusters"`. The one-step approximation is derived from the ordinary generalized estimating equations, so it is available only for fits with `method = "gee"`; for the other estimation methods, use `method = "exact"`.

When deleting a cluster or an observation leaves coefficients that cannot be estimated, for example the only cluster at a level of a factor, the leverage matrix H_i has an eigenvalue of 1 and the deletion estimate is undefined. Its values are then NA, with a warning naming the clusters or observations, and no refit is attempted for them with `method = "exact"`.

`dfbetas()` divides each column of `dfbeta()` by the standard error of the coefficient from `vcov(model, cov_type = cov_type)`. `cooks.distance()` returns

$$\frac{1}{q} (\hat{\beta} - \hat{\beta}_{(i)})^\top \hat{V}^{-1} (\hat{\beta} - \hat{\beta}_{(i)}),$$

where the coefficients are restricted to those in `parm`, q is their number and $\hat{V}$ their covariance matrix from `vcov(model, cov_type = cov_type)`. With the default `cov_type = "naive"` and all

coefficients, this is equation (4.68) of Hardin and Hilbe (2013) for clusters, and the corresponding measure of Preisser and Qaqish (1996) for observations. **glmtoolbox** uses the robust covariance matrix by default.

Value

For `hatvalues()` and `cooks.distance()`, a numeric vector with one value per cluster, named by the cluster identifier, or one value per observation, in the order and with the names given under Observation order. For `dfbeta()` and `dfbetas()`, a matrix with one row per cluster or observation, in the same order, and one column per coefficient.

Observation order

For `unit = "observations"`, the values follow the internal order of the fitted model: the complete cases, sorted by cluster and, within cluster, by repeated. They are named "`id(repeated)`" by the cluster identifier and the value of repeated, or the position within the cluster if repeated was not supplied.

References

Hardin, J.W. and Hilbe, J.M. (2013) *Generalized Estimating Equations*, 2nd Edition. Chapman and Hall/CRC, Boca Raton.

Preisser, J.S. and Qaqish, B.F. (1996) Deletion diagnostics for generalised estimating equations. *Biometrika*, **83**, 551–562.

Vanegas, L.H., Rondon, L.M. and Paula, G.A. (2023) Generalized Estimating Equations using the new R package `glmtoolbox`. *The R Journal*, **15**, 105–133.

See Also

[local_influence](#), [residuals.geer](#), [vcov.geer](#), [geewa](#), [geewa_binary](#).

Examples

```
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  data = epilepsy,
  id = id,
  repeated = visit,
  family = poisson(link = "log"),
  corstr = "exchangeable"
)
head(hatvalues(fit))
head(dfbeta(fit))
head(dfbetas(fit, cov_type = "robust"))
head(cooks.distance(fit))
head(cooks.distance(fit, unit = "observations"))
head(cooks.distance(fit, parm = "treatmentprogabide"))
```

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.

Source

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

Examples

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

local_influence	<i>Local Influence for a geer Object</i>
-----------------	--

Description

Computes local influence measures (Cook, 1986) of the clusters or the observations on the estimated regression coefficients of a fitted geer object.

Usage

```
local_influence(
  object,
  perturbation = c("cluster-weights", "observation-weights", "response"),
  type = c("total", "local"),
  parm
)
```

Arguments

object	a fitted model object of class "geer".
perturbation	character string specifying the perturbation scheme: case weights on the clusters ("cluster-weights"), case weights on the observations ("observation-weights"), or additive perturbation of the responses ("response"). See Details. Defaults to "cluster-weights".
type	character string specifying the measure: the total local influence of each cluster or observation ("total") or the absolute elements of the direction of maximum curvature ("local"). See Details. Defaults to "total".
parm	a specification of which coefficients are of interest, either a vector of numbers or a vector of names. If missing, all coefficients are used.

Details

Let $U(\beta, \omega)$ be the generalized estimating equations perturbed by ω , with $\omega = \omega_0$ giving the unperturbed equations, $\Delta = \partial U / \partial \omega^\top$ and $S = -\partial U / \partial \beta^\top$, both evaluated at $\hat{\beta}$ and ω_0 . The curvature in the direction ℓ , $\|\ell\| = 1$, is

$$C_\ell = 2|\ell^\top \Delta^\top S^{-1} \Delta \ell|.$$

With `type = "total"`, the value for cluster or observation i is the total local influence C_i , the curvature in the direction of the i -th unit vector (Lesaffre and Verbeke, 1998). With `type = "local"`, it is the absolute value of the i -th element of $\ell_{\max}$, the eigenvector of the symmetric part of $\Delta^\top S^{-1} \Delta$ with the largest absolute eigenvalue, computed by an eigendecomposition. Only the symmetric part enters C_ℓ , and S is in general not symmetric for `geewa` fits. When `parm` selects a subset of the coefficients, S^{-1} is replaced by $S^{-1} - \text{diag}(0, S_{22}^{-1})$, where S_{22} is the block of the other coefficients (Cook, 1986).

With `perturbation = "cluster-weights"`, the contribution of cluster i to the estimating equations is multiplied by ω_i . With `"observation-weights"`, the contribution of observation j of

cluster i is multiplied by ω_{ij} , that is, $D_i^\top \text{diag}(\omega_i) V_i^{-1} (y_i - \mu_i)$. With "response", the response y_{ij} is replaced by $y_{ij} + \omega_{ij} \sqrt{v_{ij}}$, where v_{ij} is the working variance of y_{ij} . Here D_i is the derivative of the mean vector of cluster i with respect to β and V_i its working covariance matrix, including the dispersion.

S is the observed derivative of the estimating equations with the association and dispersion parameters held at their estimates, including the dependence of V_i on the means, which for [geewa_binary](#) fits also enters through the odds ratios. The measures are derived from the ordinary generalized estimating equations, so they are available only for fits with `method = "gee"`.

glmtoolbox (Vanegas, Rondon and Paula, 2023) reports $\ell^\top \Delta^\top S^{-1} \Delta \ell$ without the factor 2 and the absolute value, and leaves the dispersion out of V_i , so that its total local influence for the weight perturbations is $\hat{\phi}$ times that of `local_influence()` divided by 2, while the response perturbation does not depend on these conventions. **glmtoolbox** also inverts S through a Cholesky factorization, which uses only its upper triangle, so its values differ from those of `local_influence()` when S is not symmetric, as for a correlated working structure unless the response is Gaussian with the identity link.

Value

A numeric vector with one value per cluster, named by the cluster identifier, for `perturbation = "cluster-weights"`, and one value per observation, in the order and with the names given under `Observation order`, otherwise.

Observation order

For `perturbation = "observation-weights"` and "response", the values follow the internal order of the fitted model: the complete cases, sorted by cluster and, within cluster, by repeated. They are named `"id(repeated)"` by the cluster identifier and the value of repeated, or the position within the cluster if repeated was not supplied.

References

- Cook, R.D. (1986) Assessment of local influence. *Journal of the Royal Statistical Society, Series B*, **48**, 133–169.
- Lesaffre, E. and Verbeke, G. (1998) Local influence in linear mixed models. *Biometrics*, **54**, 570–582.
- Vanegas, L.H., Rondon, L.M. and Paula, G.A. (2023) Generalized Estimating Equations using the new R package `glmtoolbox`. *The R Journal*, **15**, 105–133.

See Also

[cooks.distance.geer](#), [hatvalues.geer](#), [geewa](#), [geewa_binary](#).

Examples

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

```

    repeated = visit,
    family = poisson(link = "log"),
    corstr = "exchangeable"
  )
  head(local_influence(fit))
  head(local_influence(fit, type = "local"))
  head(local_influence(fit, perturbation = "response"))
  head(local_influence(fit, parm = "treatmentprogabide"))

```

marginaleffects-support

marginaleffects Support for geer Objects

Description

Methods that allow **marginaleffects** functions, such as `marginaleffects::predictions()`, `marginaleffects::comparisons()`, and `marginaleffects::slopes()`, to be applied to fitted `geer` objects. They are called by **marginaleffects** and not usually by the user. Inference uses the bias-corrected covariance matrix unless `vcov` requests another estimator.

Usage

```

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

## S3 method for class 'geer'
get_coef(model, ...)

## S3 method for class 'geer'
set_coef(model, coefs, ...)

## S3 method for class 'geer'
get_vcov(model, vcov = NULL, ...)

## S3 method for class 'geer'
get_predict(model, newdata = NULL, type = NULL, ...)

```

Arguments

<code>x</code>	a fitted model object of class "geer".
<code>...</code>	additional arguments passed through by marginaleffects .
<code>model</code>	a fitted model object of class "geer".
<code>coefs</code>	a named numeric vector of regression coefficients.
<code>vcov</code>	covariance specification: <code>NULL</code> or <code>TRUE</code> for the bias-corrected covariance matrix, <code>FALSE</code> for none, the name of one of the estimators described in vcov.geer , or a covariance matrix or function. Defaults to <code>NULL</code> .

newdata	optional data frame of predictor values at which predictions are computed. Defaults to NULL, which uses the data used to fit the model.
type	character string specifying the prediction scale, "response" or "link". Defaults to NULL, which is treated as "response".

Value

These methods return objects in the formats required internally by **marginaleffects**: a named coefficient vector, a modified geer object, a named covariance matrix (or NULL when vcov = FALSE), a model data frame, or a data frame with columns rowid and estimate.

See Also

[predict.geer](#), [vcov.geer](#), [coef.geer](#).

Examples

```
if (requireNamespace("marginaleffects", quietly = TRUE)) {
  data("cerebrovascular", package = "geer")
  fit <- geewa_binary(
    formula = ecg ~ treatment + factor(period),
    link = "logit",
    data = cerebrovascular,
    id = id,
    repeated = period,
    orstr = "exchangeable"
  )
  marginaleffects::avg_predictions(fit)
  marginaleffects::avg_comparisons(fit, variables = "treatment")
  marginaleffects::avg_predictions(fit, vcov = "robust")
}
```

mcar_homoscedasticity_test

Jamshidian-Jalal Test of Missing Completely at Random

Description

Tests the homogeneity of the covariance matrices across missingness patterns, a consequence of MCAR, by the procedure of Jamshidian and Jalal (2010) as implemented in **MissMech** (Jamshidian, Jalal and Jansen, 2014).

Usage

```
mcar_homoscedasticity_test(
  object,
  data = NULL,
```

```

method = c("auto", "nonparametric", "hawkins"),
imputation = c("distribution-free", "normal"),
n_imputations = 1L,
imputed_data = NULL,
min_pattern_size = 7L,
n_replicates = 10000L,
n_min = 30L,
sig_level = 0.05,
seed = 110L,
maxiter = 1000L,
tolerance = 1e-08
)

```

Arguments

object	a fitted model object of class "geer", or a numeric matrix or data frame with missing values. For a fitted model object, the response is arranged with one row per cluster and one column per occasion.
data	optional data frame containing the data used to fit object. Used only when object is a fitted model object and the data cannot be recovered from it. Defaults to NULL.
method	character string specifying the diagnostic: the modified Hawkins test ("hawkins"), the k-sample Anderson-Darling test ("nonparametric"), or the two-stage rule of Jamshidian and Jalal (2010) ("auto"). See Details. Defaults to "auto".
imputation	character string specifying how the data are completed before testing: by adding resampled complete-case residuals to the best linear predictors of the missing values ("distribution-free"), or by drawing from the conditional multivariate normal distribution ("normal"). See Details. Must not be combined with imputed_data. Defaults to "distribution-free".
n_imputations	positive integer giving the number of imputed data sets. The primary results use the first one, as in MissMech , and the results for all of them are returned in imputations. Must be 1 when imputed_data is supplied. Defaults to 1.
imputed_data	optional completed version of the matrix or data frame supplied in object, with the same dimensions and row order, no missing values, and the observed values unchanged. No imputation is then performed. Defaults to NULL.
min_pattern_size	integer, at least 2, giving the minimum number of cases for a missingness pattern to be retained. Defaults to 7, as in Jamshidian and Jalal (2010) and MissMech .
n_replicates	positive integer giving the number of simulated samples for the null distribution of the Neyman statistic in patterns with fewer than n_min cases. Defaults to 10000.
n_min	integer, at least 2, giving the pattern size from which the chi-squared reference with four degrees of freedom is used for the Neyman statistic. Defaults to 30.
sig_level	a single number strictly between 0 and 1 giving the significance level of the rule used with method = "auto". Defaults to 0.05.

seed	NULL or a single integer giving the random number seed used for imputation and for the simulated Neyman p-values. The previous random number state is restored on exit, and NULL uses the current random number stream. Defaults to 110.
maxiter	positive integer giving the maximum number of iterations of the EM algorithm used for normal-theory imputation. Defaults to 1000.
tolerance	a single positive finite number giving the relative convergence tolerance of the EM algorithm. Defaults to $1e-08$.

Details

The procedure completes the incomplete data by imputation, groups the cases by their original missingness pattern, and tests whether the completed data have a common covariance matrix across patterns. Patterns with fewer than `min_pattern_size` cases are omitted; at least two patterns, one of them with missing values, must remain.

Two tests are available. The modified Hawkins test (Hawkins, 1981) transforms the within-pattern Mahalanobis distances to variables that are Uniform(0, 1) under multivariate normality and homogeneous covariances, applies a fourth-order Neyman smooth test within each pattern, and combines the pattern p-values by Fisher's method. For patterns with fewer than `n_min` cases the null distribution of the Neyman statistic is simulated and the p-value is $(1 + b)/(n_{replicates} + 1)$, where b is the number of simulated statistics at least as large as the observed one. The nonparametric test compares the distributions of the transformed distances across patterns with the k-sample Anderson-Darling test of Scholz and Stephens (1987), which is less sensitive to nonnormality. With `method = "auto"` both tests are computed: if the Hawkins test does not reject, there is no evidence against MCAR; if it rejects but the nonparametric test does not, the rejection is attributed to nonnormality; rejection by the nonparametric test is evidence against covariance homogeneity and hence against MCAR.

Distribution-free imputation, in the spirit of Srivastava and Dolatabadi (2009), requires at least 10, and at least $2p$, complete cases among the retained rows, where p is the number of variables; otherwise normal-theory imputation is used with a warning. Because normal-theory imputation can inflate the size of the nonparametric test for nonnormal data, requesting it with `method` other than "hawkins" gives a warning. `imputed_data` is intended for matrix or data frame input, since the matrix reconstructed from a fitted model object is not returned.

The implementation follows **MissMech** (Jamshidian, Jalal and Jansen, 2014), which differs from Jamshidian and Jalal (2010) in that one imputation method serves both tests, normal-theory imputation replaces distribution-free imputation when there are too few complete cases, and the chi-squared reference is used for the Neyman statistic in patterns with at least `n_min` cases. Setting `n_min` larger than the largest pattern simulates the reference for all patterns, as in Jamshidian and Jalal (2010). This function differs from **MissMech** in two respects: the simulated Neyman p-values use $(1 + b)/(n_{replicates} + 1)$, and the Anderson-Darling p-value is interpolated from the simulated quantiles of the **kSamples** package (Scholz and Zhu, 2025) instead of the five critical values of Scholz and Stephens (1987). P-values can therefore differ from **MissMech**, particularly below 0.01. The quantiles cover upper-tail probabilities from 10^{-5} to $1 - 10^{-5}$; beyond this range the p-value is extrapolated and `nonparametric$extrapolated` is TRUE.

With `n_imputations > 1` the tests are repeated on each completed data set, holding the estimates used for imputation fixed. Jamshidian and Jalal (2010, Section 6) suggest examining the variability of the p-values and the pattern-specific results to identify patterns whose covariance differs. These results are exploratory and are not pooled.

Both tests treat the cases within a pattern as approximately independent, which Jamshidian and Jalal (2010) and Jamshidian, Jalal and Jansen (2014) found adequate for patterns with at least four cases. Known groups with different covariance matrices can cause rejection even under MCAR; conversely, supplying complete data through `imputed_data` with each group given its own artificial pattern tests covariance homogeneity across the groups (Jamshidian, Jalal and Jansen, 2014, Example 5).

The procedure ignores the marginal regression model and is a screening diagnostic: failure to reject does not establish MCAR. For longitudinal geer fits, `mcar_logistic_test` assesses whether covariates or previous responses predict missingness.

Value

An object of class `c("geer_htest", "htest")`. The standard components hold the statistic and p-value of the selected test. In addition, the object contains:

<code>tests</code>	a data frame with the statistics and p-values of the tests for the first imputation.
<code>hawkins</code>	the detailed results of the Hawkins test for the first imputation. With <code>method = "nonparametric"</code> , only the quantities used by the nonparametric test are computed and the others are NA.
<code>nonparametric</code>	the detailed results of the Anderson-Darling test for the first imputation, if computed, including the pattern contributions <code>group.statistics</code> , reported as T_i by MissMech , and the indicator extrapolated.
<code>imputations</code>	a list with the statistics and p-values (<code>tests</code>), the pattern Neyman p-values (<code>hawkins.group.p.values</code>), and the pattern Anderson-Darling contributions (<code>nonparametric.group.statistics</code>) of every imputation.
<code>n.imputations</code>	the number of imputed data sets.
<code>patterns, pattern.counts</code>	the retained missingness patterns, with 1 for a missing value, and their numbers of cases.
<code>omitted.patterns</code>	the patterns omitted by <code>min_pattern_size</code> .
<code>n, p</code>	the numbers of retained rows and of variables.
<code>imputation.requested, imputation.used</code>	the requested and the actual imputation method, "user-supplied" with <code>imputed_data</code> .
<code>complete.cases</code>	the number of retained complete cases.
<code>imputed.data</code>	the first completed data set.
<code>location, covariance</code>	the estimates used for imputation, NULL with <code>imputed_data</code> .
<code>em.iterations, em.converged</code>	the number of EM iterations and whether the EM algorithm converged, NA unless normal-theory imputation was used.
<code>method.requested, selected.test</code>	the requested diagnostic and the test giving the primary result.
<code>sig.level</code>	the significance level of the rule used with <code>method = "auto"</code> .
<code>interpretation</code>	a short interpretation of the first imputation.

References

- Hawkins, D.M. (1981) A new test for multivariate normality and homoscedasticity. *Technometrics*, **23**, 105–110.
- Jamshidian, M. and Jalal, S. (2010) Tests of homoscedasticity, normality, and missing completely at random for incomplete multivariate data. *Psychometrika*, **75**, 649–674.
- Jamshidian, M., Jalal, S. and Jansen, C. (2014) MissMech: An R package for testing homoscedasticity, multivariate normality, and missing completely at random (MCAR). *Journal of Statistical Software*, **56**, 1–31.
- Scholz, F.W. and Stephens, M.A. (1987) K-sample Anderson-Darling tests. *Journal of the American Statistical Association*, **82**, 918–924.
- Scholz, F. and Zhu, A. (2025) *kSamples: K-Sample Rank Tests and their Combinations*. R package version 1.2-12. <https://CRAN.R-project.org/package=kSamples>
- Srivastava, M.S. and Dolatabadi, M. (2009) Multiple imputation and other resampling schemes for imputing missing observations. *Journal of Multivariate Analysis*, **100**, 1919–1937.

See Also

[mcar_little_test](#), [mcar_logistic_test](#), [geewa](#), [geewa_binary](#).

Examples

```
set.seed(1)
x <- matrix(rnorm(240), ncol = 3)
x[41:60, 3] <- NA
x[61:80, 2:3] <- NA
mcar_homoscedasticity_test(x, method = "nonparametric")

## Several imputations
out <- mcar_homoscedasticity_test(x, method = "nonparametric",
                                n_imputations = 5)
out$imputations$tests
```

mcar_little_test

Little's Test of Missing Completely at Random

Description

Performs Little's (1988) test of the hypothesis that the missing values are missing completely at random (MCAR).

Usage

```
mcar_little_test(
  object,
  data = NULL,
  maxiter = 1000L,
  tolerance = 1e-08,
  reference = c("auto", "asymptotic")
)
```

Arguments

object	a fitted model object of class "geer", or a numeric matrix or data frame with one row per independent unit and one column per variable or repeated measurement. See Details.
data	optional data frame containing the data used to fit object. Used only when object is a fitted model object and the data cannot be recovered from it. Defaults to NULL.
maxiter	positive integer giving the maximum number of iterations of the EM algorithm used to estimate the multivariate normal mean and covariance matrix. Defaults to 1000.
tolerance	a single positive finite number giving the relative convergence tolerance of the EM algorithm. Defaults to 1e-08.
reference	character string specifying the reference distribution: Little's exact F reference when the data are bivariate monotone and the chi-squared reference otherwise ("auto"), or always the chi-squared reference ("asymptotic"). Defaults to "auto".

Details

Let the rows be grouped into missing-data patterns. For pattern r , let n_r be its number of rows, p_r the number of observed variables, and $\bar{y}_r$ their mean vector. With $\hat{\mu}$ and $\hat{\Sigma}$ the maximum likelihood estimates of the common mean vector and covariance matrix under a multivariate normal model, obtained by the EM algorithm, and $\tilde{\Sigma} = n\hat{\Sigma}/(n-1)$, Little's statistic is

$$d^2 = \sum_r n_r (\bar{y}_r - \hat{\mu}_r)^\top \tilde{\Sigma}_r^{-1} (\bar{y}_r - \hat{\mu}_r),$$

which under MCAR is asymptotically chi-squared with $\sum_r p_r - p$ degrees of freedom, where p is the number of variables.

When there are two variables, one fully observed, Little shows that $d^2 = (n-1)F/(n-2+F)$ with $F \sim F_{1,n-2}$. With reference = "auto" this exact reference is used in that case, provided the identity is reproduced numerically from the EM estimates; otherwise a warning is issued and the chi-squared reference is used, which usually indicates that tolerance is too loose.

Matrix or data frame input must have at least two rows and two variables, at least two observed values per variable, at least one jointly observed value for every pair of variables, and no infinite values. Rows with no observed values belong to no pattern and are removed with a warning. For a fitted model object, the response is arranged with one row per cluster and one column per repeated

index; if repeated was not supplied when fitting, the row order within each cluster defines the occasions.

The test assumes a multivariate normal model and is most appropriate for quantitative variables (Little, 1988), so a warning is issued for binary variables. It is a diagnostic for the missingness mechanism rather than for the fitted mean model (Hardin and Hilbe, 2013): a large p-value does not establish MCAR. Without missing values the statistic is 0 and the p-value 1.

Value

An object of class `c("geer_htest", "htest")`. In addition to the standard components, the object contains:

<code>missing.patterns</code>	the number of distinct missing-data patterns.
<code>n</code>	the number of independent rows or clusters tested.
<code>variables</code>	the number of variables or repeated measurements tested.
<code>iterations</code>	the number of EM iterations used.
<code>converged</code>	whether the EM algorithm converged.
<code>mean</code>	the maximum-likelihood estimate of the common mean vector.
<code>covariance</code>	the degrees-of-freedom-corrected covariance matrix used in Little's statistic.
<code>ml.covariance</code>	the uncorrected multivariate-normal maximum-likelihood covariance estimate.
<code>reference</code>	the reference distribution used for the reported p-value.
<code>asymptotic.df</code>	the degrees of freedom for the large-sample chi-squared reference.
<code>asymptotic.p.value</code>	the large-sample chi-squared p-value.
<code>exact.p.value</code>	the exact bivariate monotone p-value when available, otherwise NULL.
<code>exact.f.statistic</code>	the corresponding bivariate ANOVA F statistic when available, otherwise NULL.

References

Hardin, J.W. and Hilbe, J.M. (2013) *Generalized Estimating Equations*, 2nd Edition. Chapman and Hall/CRC, Boca Raton.

Little, R.J.A. (1988) A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, **83**, 1198–1202.

See Also

[mcar_logistic_test](#), [mcar_homoscedasticity_test](#), [geewa](#), [geewa_binary](#).

Examples

```
x <- data.frame(
  visit1 = c(2.1, 4.3, 3.2, 5.8, 7.1, 6.4, 8.2, 9.5, 10.1, 11.3),
  visit2 = c(1.5, 3.8, 2.9, 6.2, NA, 5.1, 8.9, NA, 9.4, 12.0),
  visit3 = c(3.0, 4.9, NA, 7.1, 6.3, 7.8, NA, 10.2, 11.5, 13.1)
```

```
)
mcar_little_test(x)
```

mcar_logistic_test	<i>Logistic Regression Test of Missing Completely at Random</i>
--------------------	---

Description

Tests whether the missingness of a longitudinal response depends on the previous response or on observed covariates, using a logistic regression in the spirit of Ridout (1991) and Fitzmaurice, Heath and Clifford (1996) fitted by generalized estimating equations. MCAR is defined as in Rubin (1976).

Usage

```
mcar_logistic_test(
  object,
  formula = NULL,
  data = NULL,
  orstr = c("independence", "exchangeable", "unstructured"),
  test = c("wald", "score", "working-wald", "working-score", "working-lrt"),
  cov_type = "bias-corrected",
  pmethod = c("rao-scott", "satterthwaite"),
  control = geer_control()
)
```

Arguments

object	a fitted model object of class "geer".
formula	optional one-sided formula of fully observed covariates of missingness, for example <code>~ treatment + age</code> . The previous response is added automatically and must not be included. Terms aliased with the occasion effects are dropped and reported in <code>dropped_covariates</code> . Defaults to <code>NULL</code> , which uses the right-hand side of the mean model of object.
data	optional data frame containing the data used to fit object. Used only when object is a fitted model object and the data cannot be recovered from it. Defaults to <code>NULL</code> .
orstr	character string specifying the working odds-ratio structure of the binary missingness model: "independence", "exchangeable", or "unstructured". Defaults to "independence".
test	character string specifying the hypothesis test: Wald ("wald"), generalized score ("score"), modified working Wald ("working-wald"), modified working score ("working-score"), or modified working likelihood ratio ("working-lrt"). Defaults to "wald".
cov_type	character string specifying the covariance matrix estimator used for inference: one of the estimators described in vcov.geer . Defaults to "bias-corrected".

pmethod	character string specifying the approximation to the null distribution of the modified working tests: Rao-Scott ("rao-scott") or Satterthwaite ("satterthwaite"). Ignored for the Wald and score tests. Defaults to "rao-scott".
control	list of control parameters as returned by <code>geer_control</code> for the binary GEE fit. Defaults to <code>geer_control()</code> .

Details

For each cluster i and occasion t , let $R_{it} = I(Y_{it} \text{ is missing})$. The transitions with $Y_{i,t-1}$ observed form the risk set, on which the model

$$\text{logit}\{P(R_{it} = 1)\} = \alpha_t + X_{it}^\top \gamma + \delta Y_{i,t-1}$$

is fitted by `geewa_binary` with `method = "gee"`, so that the missingness indicators of a cluster may be correlated. The occasion effects α_t are nuisance parameters. The component tests contains three tests, all computed by the procedure selected by `test`:

`response_history` $H_0 : \delta = 0$. Rejection indicates that missingness depends on the previous response, which is evidence against covariate-dependent MCAR. This is the primary test.

`covariates` $H_0 : \gamma = 0$. Rejection is evidence against MCAR but is compatible with covariate-dependent MCAR.

`overall` $H_0 : \gamma = 0, \delta = 0$.

The modified working likelihood ratio test requires `orstr = "independence"`. With score tests and `cov_type = "jackknife"`, the covariance component is computed from the larger model. The small-sample estimators of `vcov.geer` are evaluated at the fitted means of the smaller model, which extends their published definitions.

Under monotone dropout each cluster contributes up to its first missing response. With intermittent missingness a warning is issued, because the result is then a local transition diagnostic rather than a dropout test. Covariates in `formula` must be fully observed, and measurement rows that are absent from the data cannot be detected: when no response is missing but some clusters have no rows for some occasions, a warning is issued. The test is a diagnostic: failure to reject does not establish MCAR and does not rule out dependence on unobserved responses.

Value

An object of class `c("geer_htest", "htest")`. In addition to the standard components, the object contains:

tests	a data frame containing the response-history, covariate, and overall tests using the selected procedure.
coefficients	the coefficient table of the binary GEE missingness model, or NULL when no missing response occurs in the risk set.
model	the fitted model object of class "geer" returned by <code>geewa_binary</code> , or NULL when no missing response occurs in the risk set.
formula	the one-sided covariate formula.
dropped_covariates	the model-matrix columns omitted because they are aliased with the occasion effects or with earlier covariate columns.

missing, observed, transitions
 the numbers of missing outcomes, observed outcomes, and transitions in the risk set.

clusters the number of clusters represented in the risk set.

intermittent whether an observed response occurs after a missing response for at least one cluster.

orstr, test, cov_type, pmethod
 the association structure, testing procedure, covariance estimator, and working-test approximation used for inference.

References

- Fitzmaurice, G.M., Heath, A.F. and Clifford, P. (1996) Logistic regression models for binary panel data with attrition. *Journal of the Royal Statistical Society: Series A*, **159**, 249–263.
- Ridout, M.S. (1991) Testing for random dropouts in repeated measurement data. *Biometrics*, **47**, 1617–1619.
- Rubin, D.B. (1976) Inference and missing data. *Biometrika*, **63**, 581–592.

See Also

[mcar_little_test](#), [mcar_homoscedasticity_test](#), [geewa](#), [geewa_binary](#).

Examples

```
set.seed(1)
id <- rep(seq_len(60), each = 4)
time <- rep(seq_len(4), times = 60)
treatment <- rep(rep(c(0, 1), each = 30), each = 4)
y_complete <- 2 + 0.4 * treatment + 0.2 * time + rnorm(length(id))
y <- y_complete
for (i in seq_len(60)) {
  idx <- which(id == i)
  for (j in 2:4) {
    p <- plogis(-2.8 + 0.5 * treatment[idx[j]] +
      0.45 * y_complete[idx[j - 1]])
    if (rbinom(1, 1, p) == 1) {
      y[idx[j:4]] <- NA_real_
      break
    }
  }
}
dat <- data.frame(id, time, treatment, y)

fit <- geewa(
  formula = y ~ treatment + time,
  family = gaussian(),
  data = dat,
  id = id,
  repeated = time,
  corstr = "independence"
```

```

)
out <- mcar_logistic_test(fit, formula = ~ treatment, test = "wald")
out
out$tests

## Generalized score version of the same diagnostic
mcar_logistic_test(fit, formula = ~ treatment, test = "score")

```

model.matrix.geer

Construct Design Matrices from a geer Object

Description

Extracts the model matrix of the marginal mean model of a fitted geer object.

Usage

```

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

```

Arguments

object	a fitted model object of class "geer".
...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Details

The matrix is the one stored when the model was fitted, so it agrees with the coefficients, contrasts, and observations used in the fit, and its rows are in the internal order of the fitted model.

Value

The model matrix, with attributes "assign", which maps the columns to the terms (0 for the intercept), and, for models with factors, "contrasts".

See Also

[geewa](#), [geewa_binary](#), [model.matrix](#).

Examples

```

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

```

```

    repeated = period,
    orstr = "exchangeable"
  )
  head(model.matrix(fit))

```

plot.geer_runs_test *Plot the Residual Sign Sequence of a Runs Test*

Description

Plots the sequence of residual signs tested by [runs_test](#), as in Hardin and Hilbe (2013, Section 4.2.1), with successive runs in different colors and, under the natural ordering, the cluster boundaries.

Usage

```

## S3 method for class 'geer_runs_test'
plot(
  x,
  cluster_breaks = NULL,
  col = NULL,
  run_colors = grDevices::palette.colors(4L, "Okabe-Ito"),
  pch = 1L,
  cex = NULL,
  main = NULL,
  sub = NULL,
  xlab = NULL,
  ylab = NULL,
  ...
)

```

Arguments

x	an object of class "geer_runs_test", as returned by runs_test .
cluster_breaks	logical indicating whether dotted lines mark the boundaries between clusters. Defaults to NULL, which draws them only under the natural ordering.
col	color of the symbols. Defaults to NULL, which colors the symbols by run using run_colors.
run_colors	vector of at least three colors, recycled across successive runs. Defaults to grDevices::palette.colors(4L, "Okabe-Ito").
pch	plotting symbol, passed to points . Defaults to 1L.
cex	symbol size, passed to points . Defaults to NULL, which shrinks the symbols for long sequences.

main, sub, xlab, ylab
 character strings giving the title, subtitle, and axis labels. Defaults to NULL, which gives informative labels; use "" to suppress one.

... further named graphical parameters passed to `plot.default`, such as `ylim`. type cannot be changed.

Details

Each tested residual is plotted at +1 or -1 against its position in the tested sequence; zero residuals are excluded and their number is given in the default subtitle. The default title gives the ordering, and the default subtitle the number of runs, its null expectation, and the p-value. At least three colors are needed, because with two the color would simply repeat the sign. Cluster boundaries are meaningful only under the natural ordering, and requesting them otherwise gives a warning.

Value

Invisibly, a data frame with one row per tested residual and columns position, sign, run, and cluster.

References

Hardin, J.W. and Hilbe, J.M. (2013) *Generalized Estimating Equations*, 2nd Edition. Chapman and Hall/CRC, Boca Raton.

See Also

`runs_test`, `residuals.geer`.

Examples

```
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  repeated = visit,
  corstr = "exchangeable"
)
plot(runs_test(fit))
plot(runs_test(fit, order_by = "fitted"), ylim = c(-2, 2))
```

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 = "bias-corrected",
  se.fit = FALSE,
  ...
)
```

Arguments

object	a fitted model object of class "geer".
newdata	optional data frame containing the variables used for prediction. Defaults to NULL, which uses the data used to fit the model.
type	character string specifying the scale of the predictions: linear predictor ("link") or response ("response"). Defaults to "link".
cov_type	character string specifying the covariance matrix estimator used for the standard errors when se.fit = TRUE: one of the estimators described in vcov.geer . Defaults to "bias-corrected".
se.fit	logical indicating whether standard errors are returned. Defaults to FALSE.
...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Details

The linear predictor is computed from the model matrix of newdata, or of the fitted data, and the offset, and transformed by the inverse link for type = "response". Standard errors are obtained by the delta method. When newdata is supplied and the model has an offset, the variables of the offset must be in newdata.

Value

For se.fit = FALSE, a numeric vector of predictions. For se.fit = TRUE, a list with components:

fit	the predictions.
se.fit	their standard errors.

With newdata, the results are named by its row names; otherwise they are unnamed.

Observation order

With newdata, the predictions follow its rows, and rows with missing values give NA. Without it, they follow the internal order of the fitted model: the complete cases, sorted by cluster and, within cluster, by repeated. To obtain predictions in the order of the data, supply the data as newdata.

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,
  repeated = period,
  orstr = "exchangeable"
)
head(predict(fit))
head(predict(fit, type = "response"))

## Predictions for new data, with standard errors
pred <- predict(fit, newdata = cerebrovascular[1:5, ], type = "response",
               se.fit = TRUE)
pred
```

print.geer

Print a geer Object

Description

Prints the call, the estimated regression coefficients, and the convergence information of a fitted geer object.

Usage

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

Arguments

x	a fitted model object of class "geer".
...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

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,
  repeated = visit,
  corstr = "exchangeable"
)
print(fit)
```

<code>print.summary.geer</code>	<i>Print a summary.geer Object</i>
---------------------------------	------------------------------------

Description

Prints a `summary.geer` object: the call, the estimation method, the sample and cluster sizes, the coefficient table, the dispersion parameter, and the working association structure.

Usage

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

Arguments

<code>x</code>	an object of class <code>"summary.geer"</code> , as returned by summary.geer .
<code>...</code>	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

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,
  repeated = visit,
  corstr = "exchangeable"
)
print(summary(fit))
```

residuals.geer	<i>Residuals from a geer Object</i>
----------------	-------------------------------------

Description

Extracts residuals of different types from a fitted geer object.

Usage

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

Arguments

object	a fitted model object of class "geer".
type	character string specifying the type of residuals: raw ("working", or its alias "response"), Pearson ("pearson"), deviance ("deviance"), or cluster-level Mahalanobis ("mahalanobis") residuals. See Details. Defaults to "working".
...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Details

With prior weights w_{ij} , the Pearson and deviance residuals are

$$r_{ij}^P = \frac{y_{ij} - \hat{\mu}_{ij}}{\sqrt{\hat{\phi}V(\hat{\mu}_{ij})/w_{ij}}}, \quad r_{ij}^D = \text{sign}(y_{ij} - \hat{\mu}_{ij})\sqrt{d_{ij}/\hat{\phi}},$$

where d_{ij} is the weighted deviance contribution of the family, as in **glmtoolbox** (Vanegas, Rondon and Paula, 2023). The Mahalanobis residual of cluster i is

$$r_i^M = \frac{1}{n_i} (y_i - \hat{\mu}_i)^\top \hat{V}_i^{-1} (y_i - \hat{\mu}_i),$$

where $\hat{V}_i$ is the fitted working covariance matrix, including the dispersion. It is a nonnegative cluster-level diagnostic.

Value

A numeric vector with one value per observation, in the order given under Observation order, or, for type = "mahalanobis", one value per cluster, named by the cluster identifier.

Observation order

The values follow the internal order of the fitted model: the complete cases, sorted by cluster and, within cluster, by repeated. Unless the data were already in this order and had no missing values, assigning the values to the data, as in `data$r <- residuals(fit)`, misaligns the rows without warning. `fit$id` and `fit$repeated` are stored in the same order and can be used to align them.

References

Vanegas, L.H., Rondon, L.M. and Paula, G.A. (2023) Generalized Estimating Equations using the new R package glmtoolbox. *The R Journal*, **15**, 105–133.

See Also

[fitted.geer](#), [predict.geer](#), [runs_test](#), [geewa](#), [geewa_binary](#), [residuals](#).

Examples

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

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 placebo and active.

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.

Source

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

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 the rinse group, with levels placebo, A, and B.

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 no and yes.

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 follow-up observations (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.

Source

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)
```

runs_test

*Wald-Wolfowitz Runs Test for geer Residuals***Description**

Performs the nonparametric Wald-Wolfowitz runs test proposed by Chang (2000) for detecting non-random patterns in residuals from a fitted generalized estimating equation model.

Usage

```
runs_test(
  object,
  order_by = "natural",
  alternative = c("two.sided", "less", "greater")
)
```

Arguments

object	a fitted model object of class "geer".
order_by	character string or numeric vector specifying the order of the residuals: by cluster and repeated index ("natural"), by fitted value ("fitted"), by a covariate named in the model matrix or in the data used to fit the model, or by a numeric vector with one finite value per observation, in the order stored in object. Ties are broken by the natural order. See Details. Defaults to "natural".
alternative	character string specifying the alternative hypothesis: "two.sided", "less" (fewer runs than expected), or "greater" (more runs than expected). Defaults to "two.sided".

Details

Let n_p and n_n be the numbers of positive and negative residuals and T the number of runs in their sign sequence. Under the null hypothesis that the signs occur in random order,

$$E(T) = \frac{2n_p n_n}{n_p + n_n} + 1, \quad V(T) = \frac{2n_p n_n (2n_p n_n - n_p - n_n)}{(n_p + n_n)^2 (n_p + n_n - 1)},$$

and $Z = (T - E(T))/\sqrt{V(T)}$ is compared with the standard normal distribution without continuity correction, as in Chang (2000) and Hardin and Hilbe (2013). Chang (2000) recommends the approximation only when $n_p > 15$ and $n_n > 15$; otherwise a warning is issued.

alternative = "less" tests against too few runs, as produced by clustered signs, an unmodeled trend, or omitted structure, and "greater" against too many runs, that is, alternating signs. Chang (2000) rejects for extreme Z in either direction, which corresponds to the two-sided default. Hardin and Hilbe (2013) report one-sided p-values in the direction of the observed Z .

Only the signs of the residuals are used. The working, Pearson, and deviance residuals have the same signs, so the test does not depend on the residual type. Zero residuals are omitted and counted in the zero component. Prior weights are ignored: each observation contributes one sign.

A character `order_by` that is not "natural", "fitted", or a model matrix column is looked up in the data used to fit the model, so a covariate omitted from the model can be used. The model frame is rebuilt from the call and checked against the stored clusters and occasions; an error is signaled if the variable adds missing values or the check fails, in which case an aligned numeric vector can be supplied instead. Factor, character, and logical variables are ordered by their codes, and date-time variables by their numeric values.

Value

An object of class `c("geer_runs_test", "geer_htest", "htest")`, which can be plotted with `plot.geer_runs_test`. The standard components hold Z (statistic), n_p and n_n (parameter), T (estimate), and $E(T)$ (`null.value`). In addition, the object contains:

<code>runs</code>	the observed number of runs, T .
<code>expected_runs</code>	the null expectation $E(T)$.
<code>variance_runs</code>	the null variance $V(T)$.
<code>positive</code>	the number of positive residuals, n_p .
<code>negative</code>	the number of negative residuals, n_n .
<code>zero</code>	the number of zero residuals omitted from the test.
<code>nonzero</code>	the number of residuals used in the test.
<code>order_by</code>	a description of the ordering used.
<code>natural_order</code>	a logical value indicating whether the natural cluster/repeated ordering was used.
<code>signs</code>	the sign sequence whose runs were counted, in the order tested and with zero residuals removed.
<code>cluster</code>	the cluster identifier of each element of signs.

Choice of ordering

The null distribution does not depend on the ordering; the ordering determines the alternative. Ordering by a variable gives power against signs that depend on that variable:

"natural" signs that are grouped within clusters or drift over time;

"fitted" signs that depend on the fitted mean, suggesting an inadequate link or variance function;

a model covariate a misspecified functional form of that covariate;

another covariate a wrongly omitted covariate;

a numeric vector whatever the vector indexes, such as batch or center.

Chang (2000) and Hardin and Hilbe (2013, Section 4.2.1) use the natural ordering, and Hardin and Hilbe also order by a covariate; ordering by fitted values or by a supplied vector is an extension.

Under the natural ordering, too few runs can arise from positive within-cluster association as well as from a misspecified mean, so a rejection does not identify the cause. Orderings that vary within clusters interleave the clusters and make a rejection more readily attributable to the mean structure. A cluster-level covariate, such as treatment arm, keeps the clusters together and behaves like the natural ordering.

Ties in the ordering are broken by the natural order, so a heavily tied ordering, such as fitted values from few covariate patterns or a factor with few levels, approaches the natural ordering.

Hardin and Hilbe (2013, Section 4.2.1) print $E(T) = 36.32$ for their second example under the natural ordering; the correct value is $2(22)(78)/100 + 1 = 35.32$, which gives $Z = 0.494$ instead of $Z = 0.200$.

References

Chang, Y.-C. (2000) Residuals analysis of the generalized linear models for longitudinal data. *Statistics in Medicine*, **19**, 1277–1293.

Hardin, J.W. and Hilbe, J.M. (2013) *Generalized Estimating Equations*, 2nd Edition. Chapman and Hall/CRC, Boca Raton.

See Also

`plot.geer_runs_test`, `residuals.geer`, `fitted.geer`, `predict.geer`.

Examples

```
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  repeated = visit,
  corstr = "exchangeable"
)
runs_test(fit)
runs_test(fit, order_by = "fitted")
runs_test(fit, order_by = "lnage")
runs_test(fit, alternative = "less")
```

Description

Performs stepwise model selection for a fitted `geer` object, moving at each step to the single-term addition or deletion that most reduces a quasi-likelihood or Gaussian pseudolikelihood information criterion.

Usage

```
step_criterion(
  object,
  scope,
  direction = c("backward", "forward", "both"),
  criterion = c("QICu", "QIC", "QICC", "AGPC", "SGPC"),
  cov_type = "robust",
  steps = 1000L,
  trace = FALSE
)
```

Arguments

object	a fitted model object of class "geer".
scope	optional formula, or list with components upper and lower (both formulas), defining the range of models examined in the stepwise search. See Details. If omitted, the search is restricted to the terms of the initial model.
direction	character string specifying the direction of the search: "backward", "forward", or "both". Defaults to "backward".
criterion	character string specifying the information criterion minimized in the search: "QICu", "QIC", "QICC", "AGPC", or "SGPC". See Details. Defaults to "QICu".
cov_type	character string specifying the covariance matrix estimator used in the penalty terms of "QIC" and "QICC": one of the estimators described in vcov.geer . It is ignored for the other criteria. Defaults to "robust", which gives the classical definitions.
steps	nonnegative integer giving the maximum number of steps. Defaults to 1000.
trace	logical indicating whether the candidate table of each step is printed during the search. Defaults to FALSE.

Details

The criteria are those of [geecriteria](#) that are intended for comparing mean models: QIC and QICu (Pan, 2001), QICC (Hardin and Hilbe, 2013), and AGPC and SGPC (Carey and Wang, 2011; Fu, Hao and Wang, 2018), as used for variable selection by **glmtoolbox** (Vanegas, Rondon and Paula, 2023). The criteria that [geecriteria](#) provides for selecting the working association structure, such as CIC, are not accepted.

The initial model must contain every term of `scope$lower`, and `scope$upper` every term of the initial model. At each step every candidate model, obtained by adding a term of `scope$upper` or deleting a term not in `scope$lower`, is refitted to the rows used by `object`, as described in [add1.geer](#). The association and dispersion parameters are re-estimated for each candidate. With `direction = "backward"` only deletions, with `"forward"` only additions, and with `"both"` additions and deletions together are considered. The candidate with the smallest value of the criterion is selected, the first in the order of the terms of the scope in case of a tie, and the search moves to it only if its value is smaller than that of the current model by more than `sqrt(.Machine$double.eps) * max(1, abs(value))`, so that differences within the convergence tolerance of the refits do not

decide a move. Moves respect marginality. The search stops when no candidate improves the criterion or after steps steps. Because each move decreases the criterion, no model is visited twice. A candidate whose criterion cannot be evaluated is skipped.

Each step refits every candidate model, but a model already fitted during the search, such as the model before the previous move when `direction = "both"`, is not refitted. With `cov_type = "jackknife"` and `criterion = "QIC"` or `"QICC"`, every candidate model is in addition refitted once per cluster, which can be slow.

The quasi-likelihood in QIC, QICu, and QICC diverges when a fitted mean reaches the boundary of its support, so these criteria should not be used when only some of the candidate models reach the boundary. Refits of such candidates warn about fitted probabilities or rates near the boundary. See [geecriteria](#).

Value

The selected model, a fitted model object of class `"geer"` with an additional component `anova` of class `c("geer_anova", "anova", "data.frame")` with one row for the initial model and one for each step, whose row names give the term added or removed, and columns `Step` (NA for the initial model), `Df`, and the value of the criterion, named after it, and an additional component `candidates`, a list with one table of class `c("geer_anova", "anova", "data.frame")` per step. Each table has one row for the current model (`<none>`) and one for each candidate, sorted by the criterion, and columns `Df` and the value of the criterion. The last table is that of the step at which the search stopped, unless it stopped at the step limit.

References

- Carey, V.J. and Wang, Y.G. (2011) Working covariance model selection for generalized estimating equations. *Statistics in Medicine*, **30**, 3117–3124.
- Fu, L., Hao, Y. and Wang, Y.G. (2018) Working correlation structure selection in generalized estimating equations. *Computational Statistics*, **33**, 983–996.
- Hardin, J.W. and Hilbe, J.M. (2013) *Generalized Estimating Equations*, 2nd Edition. Chapman and Hall/CRC, Boca Raton.
- Pan, W. (2001) Akaike's information criterion in generalized estimating equations. *Biometrics*, **57**, 120–125.
- Vanegas, L.H., Rondon, L.M. and Paula, G.A. (2023) Generalized Estimating Equations using the new R package `glmtoolbox`. *The R Journal*, **15**, 105–133.

See Also

[step_p](#), [geecriteria](#), [add1.geer](#), [drop1.geer](#), [geewa](#), [geewa_binary](#).

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", ]
fit <- geewa_binary(
  formula = status ~ baseline + treatment + gender + visit + age,
  link = "probit",
  data = respiratory2,
```

```

    id = id,
    repeated = visit,
    method = "pgee-jeffreys"
  )
  ## Backward elimination by QICu
  step_criterion(fit)

  ## Bidirectional selection by QIC within a scope
  step_criterion(
    fit,
    scope = list(
      lower = ~ baseline + treatment,
      upper = ~ (baseline + treatment + gender + visit + age)^2
    ),
    direction = "both",
    criterion = "QIC"
  )

```

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 = "bias-corrected",
  pmethod = c("rao-scott", "satterthwaite"),
  steps = 1000L
)

```

Arguments

<code>object</code>	a fitted model object of class <code>"geer"</code> .
<code>scope</code>	optional formula, or list with components <code>upper</code> and <code>lower</code> (both formulas), defining the range of models examined in the stepwise search. See Details. If omitted, the search is restricted to the terms of the initial model.

direction	character string specifying the direction of the search: "backward", "forward", or "both". Defaults to "backward".
p_enter	a single number strictly between 0 and 1 giving the p-value threshold for adding a term. It must not exceed p_remove when direction = "both". Defaults to 0.15.
p_remove	a single number strictly between 0 and 1 giving the p-value threshold for removing a term. Defaults to 0.15.
test	character string specifying the hypothesis test: Wald ("wald"), generalized score ("score"), modified working Wald ("working-wald"), modified working score ("working-score"), or modified working likelihood ratio ("working-lrt"). Defaults to "wald".
cov_type	character string specifying the covariance matrix estimator used for inference: one of the estimators described in vcov.geer . Defaults to "bias-corrected".
pmethod	character string specifying the approximation to the null distribution of the modified working tests: Rao-Scott ("rao-scott") or Satterthwaite ("satterthwaite"). Ignored for the Wald and score tests. Defaults to "rao-scott".
steps	nonnegative integer giving the maximum number of steps. Defaults to 1000.

Details

`step_p()` repeatedly calls [add1.geer](#) and [drop1.geer](#). The terms of `scope$lower` are always kept, and `scope$upper` defines the largest model considered. A single formula is taken as upper, with an empty lower. The initial model must contain every term of `scope$lower`, and `scope$upper` every term of the initial model. Moves respect marginality. With `direction = "both"`, a deletion is tried first at each step and an addition only if no deletion qualifies. The search stops when no move meets its p-value threshold, after `steps` steps, or when a move would return to a model already visited.

The tests are described in [anova.geer](#). The CIC column is computed with `cov_type`, so `cov_type = "jackknife"` refits every candidate model once per cluster and can be slow. Candidate models are refitted to the rows used by `object`, as described in [add1.geer](#).

Value

The selected model, a fitted model object of class "geer" with an additional component `anova` of class `c("geer_anova", "anova", "data.frame")` with one row for the initial model and one for each step, whose row names give the term added or removed, and columns Step (NA for the initial model), Df, Chi, Pr(>Chi), and CIC.

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

[step_criterion](#), [add1.geer](#), [drop1.geer](#), [anova.geer](#), [geecriteria](#), [geewa](#), [geewa_binary](#).

Examples

```
data("respiratory", package = "geer")
respiratory2 <- respiratory[respiratory$center == "C2", ]
fit <- geewa_binary(
  formula = status ~ (baseline + treatment + gender + visit + age)^2,
  link = "probit",
  data = respiratory2,
  id = id,
  repeated = visit,
  method = "pgee-jeffreys"
)
## Backward elimination
step_p(fit, p_remove = 0.10)

## Bidirectional selection within a scope
step_p(
  fit,
  scope = list(
    lower = ~ baseline + treatment,
    upper = ~ (baseline + treatment + gender + visit + age)^2
  ),
  direction = "both",
  p_enter = 0.10,
  test = "score",
  cov_type = "robust"
)
```

summary.geer

Summarize a geer Object

Description

Summarizes a fitted geer object, with a table of the coefficients, their standard errors, z-statistics, and p-values.

Usage

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

Arguments

object	a fitted model object of class "geer".
cov_type	character string specifying the covariance matrix estimator used for the standard errors, z-statistics, and p-values: one of the estimators described in vcov.geer . Defaults to "bias-corrected".
...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

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.
obs_no	the number of observations used in the fit.
clusters_no	the number of clusters used in the fit.
min_cluster_size	the smallest observed cluster size.
max_cluster_size	the largest observed cluster size.
iter	the number of iterations used.
converged	logical indicating whether the algorithm converged.
phi	the estimated or fixed dispersion parameter, 1 for geewa_binary fits.
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,
  repeated = visit,
  corstr = "exchangeable"
)
summary(fit)
summary(fit, cov_type = "robust")
```

tidy.geer

*Tidy a geer Object***Description**

Summarizes the regression coefficients of a fitted geer object in a data frame with one row per coefficient, following the conventions of **broom** (Robinson, Hayes and Couch, 2024).

Usage

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

Arguments

<code>x</code>	a fitted model object of class "geer".
<code>conf.int</code>	logical indicating whether Wald confidence limits are included. Defaults to FALSE.
<code>conf.level</code>	a single number strictly between 0 and 1 specifying the confidence level. Defaults to 0.95.
<code>exponentiate</code>	logical indicating whether the estimates and confidence limits are exponentiated, for example to obtain odds ratios. Defaults to FALSE.
<code>cov_type</code>	character string specifying the covariance matrix estimator used for the standard errors, z-statistics, and confidence intervals: one of the estimators described in vcov.geer . Defaults to "bias-corrected".
<code>...</code>	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Value

A data frame, a tibble if **tibble** is installed, with columns:

<code>term</code>	the name of the coefficient.
<code>estimate</code>	the estimate, exponentiated if <code>exponentiate = TRUE</code> .
<code>std.error</code>	the standard error, on the original scale.
<code>statistic</code>	the Wald z-statistic, on the original scale.
<code>p.value</code>	the two-sided p-value.
<code>conf.low, conf.high</code>	the confidence limits, if <code>conf.int = TRUE</code> .

References

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

See Also

[glance.geer](#), [vcov.geer](#), [confint.geer](#), [geewa](#), [geewa_binary](#).

Examples

```
data("epilepsy", package = "geer")
fit <- geewa(
  formula = seizures ~ treatment + lnbaseline + lnage,
  family = poisson(link = "log"),
  data = epilepsy,
  id = id,
  repeated = visit,
  corstr = "exchangeable"
)
tidy(fit)
tidy(fit, conf.int = TRUE, exponentiate = TRUE)
tidy(fit, cov_type = "robust")
```

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 = "bias-corrected", ...)
```

Arguments

object	a fitted model object of class "geer".
cov_type	character string specifying the covariance matrix estimator used for inference: "bias-corrected", "robust", "df-adjusted", "jackknife", "naive", "kauermann-carroll", "mancl-derouen", "ford-westgate", "fay-graubard", "fan-zhang-zhang", "pan", "gosho-sato-takeuchi", "wang-long", or "westgate-burchett". See Details. Defaults to "bias-corrected".
...	not used. Supplying any argument is an error, so that a misspelled argument name is reported rather than silently ignored.

Details

Let K be the number of clusters, p the number of regression parameters, D_i the derivative of the mean vector of cluster i with respect to the regression parameters, V_i its working covariance matrix, e_i its vector of response residuals, $\Omega = \sum_i D_i^\top V_i^{-1} D_i$, $U_i = D_i^\top V_i^{-1} e_i$, and $H_i = D_i \Omega^{-1} D_i^\top V_i^{-1}$. The covariance estimator is selected by `cov_type`:

"bias-corrected" the bias-corrected sandwich estimator of Morel, Bokossa and Neerchal (2003).

"robust" the sandwich estimator of Liang and Zeger (1986), $\Omega^{-1} M \Omega^{-1}$ with $M = \sum_i U_i U_i^\top$.

"df-adjusted" the sandwich estimator with the degrees-of-freedom adjustment of MacKinnon and White (1985).

"jackknife" the leave-one-cluster-out jackknife estimator.

"naive" the model-based estimator of Liang and Zeger (1986).

"kauermann-carroll" Kauermann and Carroll (2001): the robust estimator with e_i replaced by $(I - H_i)^{-1/2} e_i$, using the principal matrix square root.

"mancl-derouen" Mancl and DeRouen (2001): the robust estimator with e_i replaced by $(I - H_i)^{-1} e_i$.

"ford-westgate" Ford and Westgate (2017): the average of the Kauermann-Carroll and Mancl-DeRouen estimators.

"fay-graubard" Fay and Graubard (2001): the robust estimator with U_i replaced by $F_i U_i$, where F_i is diagonal with entries $\{1 - \min(0.75, Q_{jj})\}^{-1/2}$ and $Q = D_i^\top V_i^{-1} D_i \Omega^{-1}$.

"fan-zhang-zhang" Fan, Zhang and Zhang (2013): the robust estimator with $e_i e_i^\top$ replaced by $e_i e_i^\top - D_i \Omega^{-1} (\sum_{j \neq i} U_j U_j^\top) \Omega^{-1} D_i^\top$, pre- and post-multiplied by $(I - H_i)^{-1}$.

"pan" Pan (2001): the residual outer products are pooled over clusters after standardization by $A_i = \text{diag}(V_i)$, so that $e_i e_i^\top$ is replaced by $A_i^{1/2} C A_i^{1/2}$ with $C = K^{-1} \sum_j A_j^{-1/2} e_j e_j^\top A_j^{-1/2}$.

"gosho-sato-takeuchi" Gosho, Sato and Takeuchi (2014): the Pan estimator multiplied by $K/(K - p)$.

"wang-long" Wang and Long (2011): the Pan estimator with e_j replaced by $(I - H_j)^{-1} e_j$ in C .

"westgate-burchett" Westgate and Burchett (2016): the Pan estimator with e_j replaced by $(I - H_j)^{-1/2} e_j$ in C . With $(I - H_j)^{-1} e_j$ instead, the Mancl-DeRouen version of this estimator, it coincides with the Wang-Long estimator.

The estimators are evaluated at the estimates of the fitted model, whatever the estimation method, and for any family and link, with D_i and V_i as used by `geewa` or `geewa_binary`. In the Pan, Gosho-Sato-Takeuchi, Wang-Long, and Westgate-Burchett estimators, A_i contains the scale parameter and the prior weights, so that weighted fits are standardized by the marginal variances they assume. The scale parameter cancels in the sandwich. With unit prior weights and the logit link, the nine small-sample estimators reproduce those of the same name in the `geessbin` package, which standardizes by $\mu(1 - \mu)$.

The bias-corrected, degrees-of-freedom adjusted, and Gosho-Sato-Takeuchi estimators require more clusters than regression parameters: otherwise the bias-corrected estimator is NA and the other two give an error. The Pan, Gosho-Sato-Takeuchi, Wang-Long, and Westgate-Burchett estimators pool the clusters and require a balanced design, in which every cluster observes each repeated position once: otherwise they give an error.

The score-type tests of `anova.geer`, `add1.geer`, `drop1.geer`, `step_p`, and `mcar_logistic_test` evaluate the selected estimator at the fit of the smaller model, with D_i and V_i of the larger model. For the nine small-sample estimators this is an extension of the published estimators, which are defined at the full-model fit. The jackknife estimator is the exception: score-type tests use the jackknife estimator of the larger model.

With K clusters and $\hat{\beta}_{(i)}$ the estimate obtained after deleting cluster i , the jackknife estimator is

$$\frac{K-1}{K} \sum_{i=1}^K (\hat{\beta}_{(i)} - \bar{\beta})(\hat{\beta}_{(i)} - \bar{\beta})^\top, \quad \bar{\beta} = \frac{1}{K} \sum_{i=1}^K \hat{\beta}_{(i)}.$$

Each $\hat{\beta}_{(i)}$ is a full refit by the original estimation method, not a one-step approximation, with the working association parameters held at their full-data estimates. The dispersion is re-estimated if it was estimated in the original fit. For the two-stage methods ("bcgee-*, "opgee-jeffreys", and "hpgee-jeffreys") the first stage is repeated in each refit. At least two clusters are required, and an error is signaled if a refit does not converge. The jackknife and the nine small-sample estimates are cached in the fitted object.

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

- Fan, C., Zhang, D. and Zhang, C.H. (2013) A comparison of bias-corrected covariance estimators for generalized estimating equations. *Journal of Biopharmaceutical Statistics*, **23**, 1172–1187.
- Fay, M.P. and Graubard, B.I. (2001) Small-sample adjustments for Wald-type tests using sandwich estimators. *Biometrics*, **57**, 1198–1206.
- Ford, W.P. and Westgate, P.M. (2017) Improved standard error estimator for maintaining the validity of inference in cluster randomized trials with a small number of clusters. *Biometrical Journal*, **59**, 478–495.
- Gosho, M., Sato, T. and Takeuchi, H. (2014) Robust covariance estimator for small-sample adjustment in the generalized estimating equations: a simulation study. *Science Journal of Applied Mathematics and Statistics*, **2**, 20–25.
- Kauermann, G. and Carroll, R.J. (2001) A note on the efficiency of sandwich covariance matrix estimation. *Journal of the American Statistical Association*, **96**, 1387–1396.
- Liang, K.Y. and Zeger, S.L. (1986) Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 13–22.
- MacKinnon, J.G. and White, H. (1985) Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, **29**, 305–325.
- Mancl, L.A. and DeRouen, T.A. (2001) A covariance estimator for GEE with improved small-sample properties. *Biometrics*, **57**, 126–134.
- 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.
- Pan, W. (2001) On the robust variance estimator in generalised estimating equations. *Biometrika*, **88**, 901–906.

Wang, M. and Long, Q. (2011) Modified robust variance estimator for generalized estimating equations with improved small-sample performance. *Statistics in Medicine*, **30**, 1278–1291.

Westgate, P.M. and Burchett, W.W. (2016) Improving power in small-sample longitudinal studies when using generalized estimating equations. *Statistics in Medicine*, **35**, 3733–3744.

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,
  repeated = period,
  orstr = "exchangeable"
)
vcov(fit)
vcov(fit, cov_type = "robust")
vcov(fit, cov_type = "kauermann-carroll")
```

Index

- * **datasets**
 - cerebrovascular, [7](#)
 - cholecystectomy, [8](#)
 - depression, [11](#)
 - epilepsy, [14](#)
 - leprosy, [43](#)
 - respiratory, [65](#)
 - rinse, [66](#)
- * **package**
 - geer-package, [3](#)
- add1.geer, [3](#), [3](#), [6](#), [7](#), [70–73](#), [79](#)
- anova.geer, [3](#), [4](#), [5](#), [5](#), [20](#), [26](#), [31](#), [36](#), [37](#), [73](#), [79](#)
- brglm_control, [23](#), [28](#)
- brglmFit, [23](#), [28](#)
- cerebrovascular, [7](#)
- cholecystectomy, [8](#)
- coef.geer, [9](#), [11](#), [47](#), [62](#), [79](#), [80](#)
- confint.geer, [3](#), [10](#), [10](#), [77](#), [80](#)
- cooks.distance.geer, [45](#)
- cooks.distance.geer (hatvalues.geer), [40](#)
- cut, [36](#)
- depression, [11](#)
- dfbeta.geer (hatvalues.geer), [40](#)
- dfbetas.geer (hatvalues.geer), [40](#)
- do.call, [19](#)
- drop1.geer, [7](#), [71–73](#), [79](#)
- drop1.geer (add1.geer), [3](#)
- emm_basis.geer (emmeans-support), [12](#)
- emmeans-support, [12](#)
- epilepsy, [14](#)
- family, [25](#), [29](#), [75](#)
- fitted.geer, [15](#), [61](#), [64](#), [69](#)
- frechet_bounds_cor, [3](#), [16](#)
- geecriteria, [3](#), [5](#), [7](#), [17](#), [26](#), [31](#), [32](#), [70](#), [71](#), [73](#)
- geer (geer-package), [3](#)
- geer-package, [3](#)
- geer_control, [3](#), [21](#), [23](#), [25](#), [26](#), [28–31](#), [55](#)
- geewa, [3](#), [5](#), [7](#), [13](#), [16](#), [17](#), [20](#), [21](#), [22](#), [25](#), [29–32](#), [34](#), [35](#), [37](#), [39](#), [42](#), [44](#), [45](#), [51](#), [53](#), [56](#), [57](#), [61](#), [64](#), [71](#), [73](#), [77](#), [78](#)
- geewa_binary, [3](#), [5](#), [7](#), [13](#), [20](#), [21](#), [26](#), [27](#), [30](#), [32](#), [34](#), [35](#), [37](#), [39](#), [42](#), [45](#), [51](#), [53](#), [55–57](#), [61](#), [64](#), [71](#), [73](#), [75](#), [77](#), [78](#)
- get_coef.geer (marginaleffects-support), [46](#)
- get_data.geer (marginaleffects-support), [46](#)
- get_predict.geer (marginaleffects-support), [46](#)
- get_vcov.geer (marginaleffects-support), [46](#)
- glance.geer, [3](#), [20](#), [31](#), [75](#), [77](#)
- gof_residual_test, [3](#), [33](#), [37–39](#)
- gof_score_test, [3](#), [35](#), [35](#), [38](#), [39](#)
- gof_tests, [3](#), [35](#), [37](#), [38](#)
- hatvalues.geer, [3](#), [34](#), [35](#), [40](#), [45](#)
- leprosy, [43](#)
- local_influence, [3](#), [42](#), [44](#)
- marginaleffects-support, [46](#)
- mcar_homoscedasticity_test, [3](#), [47](#), [53](#), [56](#)
- mcar_little_test, [3](#), [51](#), [51](#), [56](#)
- mcar_logistic_test, [3](#), [50](#), [51](#), [53](#), [54](#), [79](#)
- model.frame, [24](#), [29](#)
- model.matrix, [57](#)
- model.matrix.geer, [57](#)
- plot.default, [59](#)
- plot.geer_runs_test, [58](#), [68](#), [69](#)
- points, [58](#)
- predict, [61](#)

`predict.geer`, [3](#), [15](#), [47](#), [60](#), [64](#), [69](#)
`print.geer`, [61](#), [62](#)
`print.summary.geer`, [62](#), [75](#)

`recover_data.geer (emmeans-support)`, [12](#)
`residuals`, [64](#)
`residuals.geer`, [3](#), [15](#), [42](#), [59](#), [61](#), [63](#), [69](#)
`respiratory`, [65](#)
`rinse`, [66](#)
`runs_test`, [3](#), [26](#), [31](#), [58](#), [59](#), [64](#), [67](#)

`set_coef.geer`
 (`marginaleffects-support`), [46](#)
`step_criterion`, [3](#), [20](#), [69](#), [73](#)
`step_p`, [3](#), [5](#), [7](#), [20](#), [26](#), [31](#), [71](#), [72](#), [79](#)
`summary.geer`, [3](#), [10](#), [26](#), [31](#), [62](#), [74](#), [80](#)

`terms`, [25](#), [29](#)
`tidy.geer`, [3](#), [11](#), [32](#), [75](#), [76](#), [80](#)

`vcov.geer`, [3](#), [4](#), [6](#), [10](#), [11](#), [13](#), [17](#), [20](#), [26](#), [31](#),
 [34](#), [36](#), [41](#), [42](#), [46](#), [47](#), [54](#), [55](#), [60](#), [70](#),
 [73–76](#), [77](#), [77](#)