Package ‘clubSandwich’

May 14, 2019

Title     Cluster-Robust (Sandwich) Variance Estimators with Small-Sample Corrections
Version   0.3.5
Description Provides several cluster-robust variance estimators (i.e., sandwich estimators) for ordinary and weighted least squares linear regression models, including the bias-reduced linearization estimator introduced by Bell and McCaffrey (2002) <http://www.statcan.gc.ca/pub/12-001-x/2002002/article/9058-eng.pdf> and developed further by Pustejovsky and Tipton (2017) <DOI:10.1080/07350015.2016.1247004>. The package includes functions for estimating the variance-covariance matrix and for testing single- and multiple-contrast hypotheses based on Wald test statistics. Tests of single regression coefficients use Satterthwaite or saddle-point corrections. Tests of multiple-contrast hypotheses use an approximation to Hotelling's T-squared distribution. Methods are provided for a variety of fitted models, including lm(), mmlm objects, glm(), ivreg() (from package 'AER'), plm() (from package 'plm'), gls() and lme() (from 'nlme'), robu() (from 'robumeta'), and rma.uni() and rma.mv() (from 'metafor').

URL https://github.com/jepusto/clubSandwich

BugReports https://github.com/jepusto/clubSandwich/issues

Depends R (>= 3.0.0)
License GPL-3
VignetteBuilder knitr
LazyData true
Imports stats, sandwich
Suggests Formula, knitr, carData, geepack, metafor, robumeta, nlme, mmlmRev, AER, plm (>= 1.6-4), lme4, zoo, testthat, rmarkdown
RoxygenNote 6.1.1
Encoding UTF-8
Language en-US
NeedsCompilation no
Achievement Awards Demonstration program

Description

Data from a randomized trial of the Achievement Awards Demonstration program, reported in Angrist & Lavy (2009).

Usage

AchievementAwardsRCT
Format

A data frame with 16526 rows and 21 variables:

- **school_id**  Fictitious school identification number
- **school_type**  Factor identifying the school type (Arab religious, Jewish religious, Jewish secular)
- **pair**  Number of treatment pair. Note that 7 is a triple.
- **treated**  Indicator for whether school was in treatment group
- **year**  Cohort year
- **student_id**  Fictitious student identification number
- **sex**  Factor identifying student sex
- **siblings**  Number of siblings
- **immigrant**  Indicator for immigrant status
- **father_ed**  Father’s level of education
- **mother_ed**  Mother’s level of education
- **Bagrut_status**  Indicator for Bagrut attainment
- **attempted**  Number of Bagrut units attempted
- **awarded**  Number of Bagrut units awarded
- **achv_math**  Indicator for satisfaction of math requirement
- **achv_english**  Indicator for satisfaction of English requirement
- **achv_hebrew**  Indicator for satisfaction of Hebrew requirement
- **lagscore**  Lagged Bagrut score
- **qrtl**  Quartile within distribution of lagscore, calculated by cohort and sex
- **half**  Lower or upper half within distribution of lagscore, calculated by cohort and sex

Source

Angrist Data Archive

References

coef_test reports t-tests for each coefficient estimate in a fitted linear regression model, using a sandwich estimator for the standard errors and a small sample correction for the p-value. The small-sample correction is based on a Satterthwaite approximation or a saddlepoint approximation.

Usage

```r
coef_test(obj, vcov, test = "Satterthwaite", coefs = "All", p_values = TRUE, ...)
```

Arguments

- `obj`: Fitted model for which to calculate t-tests.
- `vcov`: Variance covariance matrix estimated using `vcovCR` or a character string specifying which small-sample adjustment should be used to calculate the variance-covariance.
- `test`: Character vector specifying which small-sample corrections to calculate. "z" returns a z test (i.e., using a standard normal reference distribution). "naive-t" returns a t test with m - 1 degrees of freedom. "Satterthwaite" returns a Satterthwaite correction. "saddlepoint" returns a saddlepoint correction. Default is "Satterthwaite".
- `coefs`: Character, integer, or logical vector specifying which coefficients should be tested. The default value "All" will test all estimated coefficients.
- `p_values`: Logical indicating whether to report p-values. The default value is `TRUE`.
- `...`: Further arguments passed to `vcovCR`, which are only needed if `vcov` is a character string.

Value

A data frame containing estimated regression coefficients, standard errors, and test results. For the Satterthwaite approximation, degrees of freedom and a p-value are reported. For the saddlepoint approximation, the saddlepoint and a p-value are reported.

See Also

`vcovCR`
Examples

data("Produc", package = "plm")
lm_individual <- lm(log(gsp) ~ 0 + state + log(pcap) + log(pc) + log(emp) + unemp, data = Produc)
individual_index <- grep("state", names(coef(lm_individual)))
coef_test(lm_individual, vcov = "CR2", cluster = Produc$state, coefs = individual_index)

V_CR2 <- vcovCR(lm_individual, cluster = Produc$state, type = "CR2")
coef_test(lm_individual, vcov = V_CR2, coefs = individual_index)

Description

conf_int reports confidence intervals for each coefficient estimate in a fitted linear regression model, using a sandwich estimator for the standard errors and a small sample correction for the critical values. The small-sample correction is based on a Satterthwaite approximation.

Usage

conf_int(obj, vcov, level = 0.95, test = "Satterthwaite",
coefs = "All", ...)

Arguments

obj Fitted model for which to calculate confidence intervals.
vcov Variance covariance matrix estimated using vcovCR or a character string specifying which small-sample adjustment should be used to calculate the variance-covariance.
level Desired coverage level for confidence intervals.
test Character vector specifying which small-sample corrections to calculate. "z" returns a z test (i.e., using a standard normal reference distribution). "naive-t" returns a t test with m -1 degrees of freedom. "Satterthwaite" returns a Satterthwaite correction. "saddlepoint" returns a saddlepoint correction. Default is "Satterthwaite".
coefs Character, integer, or logical vector specifying which coefficients should be tested. The default value "All" will test all estimated coefficients.
... Further arguments passed to vcovCR, which are only needed if vcov is a character string.

Value

A data frame containing estimated regression coefficients, standard errors, and confidence intervals.
See Also

vcovCR

Examples

data("Produc", package = "plm")
lm_individual <- lm(log(gsp) ~ 0 + state + log(pcap) + log(pc) + log(emp) + unemp, data = Produc)
individual_index <- grep("state", names(coef(lm_individual)))
conf_int(lm_individual, vcov = "CR2", cluster = Produc$state, coefs = individual_index)

V_CR2 <- vcovCR(lm_individual, cluster = Produc$state, type = "CR2")
conf_int(lm_individual, vcov = V_CR2, level = .99, coefs = individual_index)

dropoutPrevention  Dropout prevention/intervention program effects

Description

A dataset containing estimated effect sizes, variances, and covariates from a meta-analysis of dropout prevention/intervention program effects, conducted by Wilson et al. (2011). Missing observations were imputed.

Usage

dropoutPrevention

Format

A data frame with 385 rows and 18 variables:

LOR1  log-odds ratio measuring the intervention effect
varLOR  estimated sampling variance of the log-odds ratio
studyID  unique identifier for each study
studySample  unique identifier for each sample within a study
study_design  study design (randomized, matched, or non-randomized and unmatched)
outcome  outcome measure for the intervention effect is estimated (school dropout, school enrollment, graduation, graduation or GED receipt)
evaluator_independence  degree of evaluator independence (independent, indirect but influential, involved in planning but not delivery, involved in delivery)
implementation_quality  level of implementation quality (clear problems, possible problems, no apparent problems)
program_site  Program delivery site (community, mixed, school classroom, school but outside of classroom)
attrition  Overall attrition (proportion)
**impute_covariance_matrix**

**Description**

`impute_covariance_matrix` calculates a block-diagonal covariance matrix, given the marginal variances, the block structure, and an assumed correlation.

**Usage**

```r
impute_covariance_matrix(vi, cluster, r,
                          return_list = identical(as.factor(cluster), sort(as.factor(cluster))))
```

**Arguments**

- `vi` Vector of variances
- `cluster` Vector indicating which effects belong to the same cluster. Effects with the same value of `cluster` will be treated as correlated.
- `r` Vector or numeric value of assume correlation(s) between effect size estimates from each study.
- `return_list` Optional logical indicating whether to return a list of matrices (with one entry per block) or the full variance-covariance matrix.

**Source**


**References**


**Value**

If `cluster` is appropriately sorted, then a list of matrices, with one entry per cluster, will be returned by default. If `cluster` is out of order, then the full variance-covariate matrix will be returned by default. The output structure can be controlled with the optional `return_list` argument.

**Examples**

```r
library(metafor)
data(SATcoaching)
V_list <- impute_covariance_matrix(vi = SATcoaching$V, cluster = SATcoaching$study, r = 0.66)
MVFE <- rma.mv(d ~ 0 + test, V = V_list, data = SATcoaching)
coef_test(MVFE, vcov = "CR2", cluster = SATcoaching$study)
```

---

**MortalityRates**

*State-level annual mortality rates by cause among 18-20 year-olds*

**Description**

A dataset containing state-level annual mortality rates for select causes of death, as well as data related to the minimum legal drinking age and alcohol consumption.

**Usage**

```r
MortalityRates
```

**Format**

A data frame with 5508 rows and 12 variables:

- `year` Year of observation
- `state` identifier for state
- `count` Number of deaths
- `pop` Population size
- `legal` Proportion of 18-20 year-old population that is legally allowed to drink
- `beertaxa` Beer taxation rate
- `beerperscap` Beer consumption per capita
- `wineperscap` Wine consumption per capita
- `spiritperscap` Spirits consumption per capita
- `totperscap` Total alcohol consumption per capita
- `mrate` Mortality rate per 10,000
- `cause` Cause of death
Source

Mastering 'Metrics data archive

References


---

**SATcoaching**

*Randomized experiments on SAT coaching*

---

**Description**

Effect sizes from studies on the effects of SAT coaching, reported in Kalaian and Raudenbush (1996)

**Usage**

SATcoaching

**Format**

A data frame with 67 rows and 11 variables:

- **study** Study identifier
- **year** Year of publication
- **test** Character string indicating whether effect size corresponds to outcome on verbal (SATV) or math (SATM) test
- **d** Effect size estimate (Standardized mean difference)
- **V** Variance of effect size estimate
- **nT** Sample size in treatment condition
- **nC** Sample size in control condition
- **study_type** Character string indicating whether study design used a matched, non-equivalent, or randomized control group
- **hrs** Hours of coaching
- **ETS** Indicator variable for Educational Testing Service
- **homework** Indicator variable for homework

**References**

vcovcr

Cluster-robust variance-covariance matrix

Description
This is a generic function, with specific methods defined for \texttt{lm}, \texttt{plm}, \texttt{glm}, \texttt{gls}, \texttt{lme}, \texttt{robu}, \texttt{rma.uni}, and \texttt{rma.mv} objects.

vcovcr returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates.

Usage

\texttt{vcovcr(obj, cluster, type, target, inverse.var, form, ...)}

## Default S3 method:
\texttt{vcovcr(obj, cluster, type, target = NULL, inverse.var = FALSE, form = "sandwich", ...)}

Arguments

\begin{itemize}
  \item \texttt{obj} Fitted model for which to calculate the variance-covariance matrix
  \item \texttt{cluster} Expression or vector indicating which observations belong to the same cluster. For some classes, the cluster will be detected automatically if not specified.
  \item \texttt{type} Character string specifying which small-sample adjustment should be used, with available options "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of \texttt{vcovcr} for further information.
  \item \texttt{target} Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If a vector, the target matrix is assumed to be diagonal. If not specified, vcovcr will attempt to infer a value.
  \item \texttt{inverse.var} Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, vcovcr will attempt to infer a value.
  \item \texttt{form} Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting \texttt{form = "meat"} will return only the meat of the sandwich and setting \texttt{form = B}, where \texttt{B} is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using \texttt{B} as the bread.
  \item \ldots Additional arguments available for some classes of objects.
\end{itemize}

Details

vcovcr returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates.

Several different small sample corrections are available, which run parallel with the "HC" corrections for heteroskedasticity-consistent variance estimators, as implemented in \texttt{vcovHC}. The "CR2"
adjustment is recommended (Pustejovsky & Tipton, 2017; Imbens & Kolesar, 2016). See Pustejovsky and Tipton (2017) and Cameron and Miller (2015) for further technical details. Available options include:

"CR0" is the original form of the sandwich estimator (Liang & Zeger, 1986), which does not make any small-sample correction.

"CR1" multiplies CR0 by \( \frac{m}{(m - 1)} \), where \( m \) is the number of clusters.

"CR1p" multiplies CR0 by \( \frac{m}{(m - p)} \), where \( m \) is the number of clusters and \( p \) is the number of covariates.

"CR1S" multiplies CR0 by \( \frac{(m - 1)}{[(m - 1)(N - p)]} \), where \( m \) is the number of clusters, \( N \) is the total number of observations, and \( p \) is the number of covariates. Some Stata commands use this correction by default.

"CR2" is the "bias-reduced linearization" adjustment proposed by Bell and McCaffrey (2002) and further developed in Pustejovsky and Tipton (2017). The adjustment is chosen so that the variance-covariance estimator is exactly unbiased under a user-specified working model.

"CR3" approximates the leave-one-cluster-out jackknife variance estimator (Bell & McCaffrey, 2002).

**Value**

An object of class c("vcovCR","clubSandwich"), which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates. The matrix has several attributes:

- **type** indicates which small-sample adjustment was used
- **cluster** contains the factor vector that defines independent clusters
- **bread** contains the bread matrix
- **v_scale** constant used in scaling the sandwich estimator
- **est_mats** contains a list of estimating matrices used to calculate the sandwich estimator
- **adjustments** contains a list of adjustment matrices used to calculate the sandwich estimator
- **target** contains the working variance-covariance model used to calculate the adjustment matrices. This is needed for calculating small-sample corrections for Wald tests.

**References**


v covCR.glm

Cluster-robust variance-covariance matrix for a glm object.

Description

v covCR returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from an glm object.

Usage

## S3 method for class 'glm'
v covCR(obj, cluster, type, target = NULL, inverse_var = NULL, form = "sandwich", ...)

Arguments

- obj: Fitted model for which to calculate the variance-covariance matrix
- cluster: Expression or vector indicating which observations belong to the same cluster. Required for glm objects.
vcovCR.gls 13

type Character string specifying which small-sample adjustment should be used, with available options "CR8", "CR1", "CR1p", "CR1s", "CR2", or "CR3". See "Details" section of vcovCR for further information.

target Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If a vector, the target matrix is assumed to be diagonal. If not specified, the target is taken to be the estimated variance function.

inverse_var Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, vcovCR will attempt to infer a value.

form Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting form = "meat" will return only the meat of the sandwich and setting form = B, where B is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using B as the bread.

... Additional arguments available for some classes of objects.

Value
An object of class c("vcovCRC", "clubSandwich"), which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also
vcovCR

Examples

data(dietox, package = "geepack")
dietox$Cu <- as.factor(dietox$Cu)
weight_fit <- glm(Weight ~ Cu * poly(Time, 3), data=dietox, family = "quasipoisson")
V.CR <- vcovCR(weight_fit, cluster = dietox$pig, type = "CR2")
coef_test(weight_fit, vcov = V.CR, test = "Satterthwaite")

vcovCR.gls  Cluster-robust variance-covariance matrix for a gls object.

Description
vcovCR returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from a gls object.

Usage
## S3 method for class 'gls'
vcovCR(obj, cluster, type, target, inverse_var,
form = "sandwich", ...)
Arguments

- **obj**: Fitted model for which to calculate the variance-covariance matrix.
- **cluster**: Optional expression or vector indicating which observations belong to the same cluster. If not specified, will be set to `getGroups(obj)`.
- **type**: Character string specifying which small-sample adjustment should be used, with available options: "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of `vcovCR` for further information.
- **target**: Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If not specified, the target is taken to be the estimated variance-covariance structure of the `gls` object.
- **inverse_var**: Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, `vcovCR` will attempt to infer a value.
- **form**: Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting `form = "meat"` will return only the meat of the sandwich and setting `form = B`, where `B` is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using `B` as the bread.
- **...**: Additional arguments available for some classes of objects.

Value

An object of class `c("vcovCR","clubSandwich")`, which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also

- `vcovCR`

Examples

```r
library(nlme)
data(Ovary, package = "nlme")
Ovary$time_int <- 1:nrow(Ovary)
lm_AR1 <- gls(follicles ~ sin(2*pi*Time) + cos(2*pi*Time), data = Ovary,
              correlation = corAR1(form = ~ time_int | Mare))
vcovCR(lm_AR1, type = "CR2")
```

---

### vcovCR.ivreg

Cluster-robust variance-covariance matrix for an ivreg object.

Description

`vcovCR` returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from an `ivreg` object.
Usage

## S3 method for class 'ivreg'
vcovCR(obj, cluster, type, target = NULL,
       inverse_var = FALSE, form = "sandwich", ...)

Arguments

- **obj**: Fitted model for which to calculate the variance-covariance matrix.
- **cluster**: Expression or vector indicating which observations belong to the same cluster. Required for ivreg objects.
- **type**: Character string specifying which small-sample adjustment should be used, with available options "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of vcovCR for further information.
- **target**: Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If a vector, the target matrix is assumed to be diagonal. If not specified, the target is taken to be an identity matrix.
- **inverse_var**: Not used for ivreg objects.
- **form**: Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting form = "meat" will return only the meat of the sandwich and setting form = B, where B is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using B as the bread.
- **...**: Additional arguments available for some classes of objects.

Value

An object of class c("vcovCR","clubSandwich"), which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also

- vcovCR

Examples

```r
library(AER)
data("CigarettesSW")
Cigs <- within(CigarettesSW, {
  rprice <- price/cpi
  rincome <- income/population/cpi
  tdiff <- (taxes - tax)/cpi
})

iv_fit <- ivreg(log(packs) ~ log(rprice) + log(rincome) |
                log(rincome) + tdiff + I(tax/cpi), data = Cigs)
vcovCR(iv_fit, cluster = Cigs$state, type = "CR2")
coef_test(iv_fit, vcov = "CR2", cluster = Cigs$state)
```
vcovCR returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from an `lm` object.

Usage

```r
## S3 method for class 'lm'
vcovCR(obj, cluster, type, target = NULL,
inverse_var = NULL, form = "sandwich", ...)
```

Arguments

- `obj`: Fitted model for which to calculate the variance-covariance matrix
- `cluster`: Expression or vector indicating which observations belong to the same cluster. Required for `lm` objects.
- `type`: Character string specifying which small-sample adjustment should be used, with available options "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of `vcovcr` for further information.
- `target`: Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If a vector, the target matrix is assumed to be diagonal. If not specified, the target is taken to be an identity matrix.
- `inverse_var`: Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, `vcovCR` will attempt to infer a value.
- `form`: Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting `form = "meat"` will return only the meat of the sandwich and setting `form = B`, where `B` is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using `B` as the bread.

Value

An object of class `c("vcovCR","clubSandwich")`, which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also

`vcovCR`
Examples

data("Produc", package = "plm")
lm_individual <- lm(log(gsp) ~ 0 + state + log(pcap) + log(pc) + log(emp) + unemp, data = Produc)
individual_index <- grepl("state", names(coef(lm_individual)))
vcovCR(lm_individual, cluster = Produc$state, type = "CR2")[individual_index,individual_index]

# compare to plm()
plm_FE <- plm::plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, data = Produc, index = c("state","year"),
effect = "individual", model = "within")
vcovCR(plm_FE, type="CR2")

vcovCR.lme

Cluster-robust variance-covariance matrix for an lme object.

Description

vcovCR returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from a lme object.

Usage

## S3 method for class 'lme'
vcovCR(obj, cluster, type, target, inverse_var,
form = "sandwich", ...)

Arguments

obj Fitted model for which to calculate the variance-covariance matrix
cluster Optional expression or vector indicating which observations belong to the same cluster. If not specified, will be set to getGroups(obj).
type Character string specifying which small-sample adjustment should be used, with available options "CR0", "CR1", "CR1p", "CR1s", "CR2", or "CR3". See "Details" section of vcovCR for further information.
target Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If not specified, the target is taken to be the estimated variance-covariance structure of the lme object.
inverse_var Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, vcovCR will attempt to infer a value.
form Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting form = "meat" will return only the meat of the sandwich and setting form = B, where B is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using B as the bread.
... Additional arguments available for some classes of objects.
Value
An object of class `c("vcovCR","clubSandwich")`, which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also
vcovCR

Examples

```r
library(nlme)
rat_weight <- lme(weight ~ Time * Diet, data=BodyWeight, ~ Time | Rat)
vcovCR(rat_weight, type = "CR2")

data(egsingle, package = "mlmRev")
math_model <- lme(math ~ year * size + female + black + hispanic,
                  random = list(~ year | schoolid, ~ 1 | childid),
                  data = egsingle)
vcovCR(math_model, type = "CR2")
```

Description
vcovCR returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from an `mlm` object.

Usage
```r
## S3 method for class 'mlm'
vcovCR(obj, cluster, type, target, inverse_var, form = "sandwich", ...)
```

Arguments
- `obj` Fitted model for which to calculate the variance-covariance matrix
- `cluster` Optional expression or vector indicating which observations belong to the same cluster. If not specified, each row of the data will be treated as a separate cluster.
- `type` Character string specifying which small-sample adjustment should be used, with available options "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of `vcovCR` for further information.
- `target` Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If not specified, the target is taken to be an identity matrix.
Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, vcovCR will attempt to infer a value.

Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting form = "meat" will return only the meat of the sandwich and setting form = B, where B is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using B as the bread.

Additional arguments available for some classes of objects.

An object of class c("vcovCR","clubSandwich"), which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

vcovCR

Examples

iris_fit <- lm(cbind(Sepal.Length, Sepal.Width) ~ Species + Petal.Length + Petal.Width, data = iris)
Vcluster <- vcovCR(iris_fit, type = "CR2")

vcovCR.plm  Cluster-robust variance-covariance matrix for a plm object.

Description

vcovCR returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from a plm object.

Usage

## S3 method for class 'plm'
vcovCR(obj, cluster, type, target, inverse_var, form = "sandwich", ignore_FE = FALSE, ...)

Arguments

obj Fitted model for which to calculate the variance-covariance matrix
cluster Optional character string, expression, or vector indicating which observations belong to the same cluster. For fixed-effect models that include individual effects or time effects (but not both), the cluster will be taken equal to the included fixed effects if not otherwise specified. Clustering on individuals can also be obtained by taking cluster = "individual" and clustering on time periods can be obtained with cluster = "time". For random-effects models, the cluster will be taken equal to the included random effect identifier if not otherwise specified.
type Character string specifying which small-sample adjustment should be used, with available options "CR8", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of vcovCR for further information.

target Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. By default, the target is taken to be an identity matrix for fixed effect models or the estimated compound-symmetric covariance matrix for random effects models.

inverse_var Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, vcovCR will attempt to infer a value.

form Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting form = "meat" will return only the meat of the sandwich and setting form = B, where B is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using B as the bread.

ignore_FE Optional logical controlling whether fixed effects are ignored when calculating small-sample adjustments in models where fixed effects are estimated through absorption.

... Additional arguments available for some classes of objects.

Value

An object of class c("vcovCR","clubSandwich"), which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also

vcovCR

Examples

library(plm)
# fixed effects
data("Produc", package = "plm")
plm_FE <- plm(log(gsp) ~ log(pcap) + log(pc) + log(emp) + unemp, 
data = Produc, index = c("state","year"),

effect = "individual", model = "within")
vcovCR(plm_FE, type="CR2")

# random effects
plm_RE <- update(plm_FE, model = "random")
vcovCR(plm_RE, type = "CR2")

# first differencing
data(Fatalities, package = "AER")
Fatalities <- within(Fatalities, {
  frate <- 10000 * fatal / pop
  drinkagec <- cut(drinkage, breaks = 18:22, include.lowest = TRUE, right = FALSE)
drinkagec <- relevel(drinkagec, ref = 4)
})
v<sub>covCR</sub>.<sub>rma.mv</sub>

```r
plm_FD <- plm(frate ~ beertax + drinkagec + miles + unemp + log(income),
    data = Fatalities, index = c("state", "year"),
    model = "fd")
vcovHC(plm_FD, method="arellano", type = "sss", cluster = "group")
vcovCR(plm_FD, type = "CR1S")
vcovCR(plm_FD, type = "CR2")
```

---

**Description**

`vcovCR` returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from a `rma.mv` object.

**Usage**

```r
## S3 method for class 'rma.mv'
vcovCR(obj, cluster, type, target, inverse_var,
    form = "sandwich", ...)
```

**Arguments**

- `obj` Fitted model for which to calculate the variance-covariance matrix.
- `cluster` Optional expression or vector indicating which observations belong to the same cluster. If not specified, will be set to the factor in the random-effects structure with the fewest distinct levels. Caveat emptor: the function does not check that the random effects are nested.
- `type` Character string specifying which small-sample adjustment should be used, with available options "CR0", "CR1", "CR1p", "CR1S", "CR2", or "CR3". See "Details" section of `vcovCR` for further information.
- `target` Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If not specified, the target is taken to be the estimated variance-covariance structure of the `rma.mv` object.
- `inverse_var` Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, `vcovCR` will attempt to infer a value.
- `form` Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting `form = "meat"` will return only the meat of the sandwich and setting `form = B`, where `B` is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using `B` as the bread.
- `...` Additional arguments available for some classes of objects.
Value

An object of class \code{c("vcovCR","clubSandwich")}, which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also

\code{vcovCR}

Examples

library(metafor)
data(hierdat, package = "robmeta")

mfor_fit <- rma.mv(effects_size ~ binge + followup + sreport + age,
                  V = var, random = list(~ 1 | esid, ~ 1 | studyid),
                  data = hierdat)
mfor_fit

mfor_CR2 <- vcovCR(mfor_fit, type = "CR2")
mfor_CR2
coefficient(mfor_fit, vcov = mfor_CR2, test = c("Satterthwaite", "saddlepoint"))

wald_test(mfor_fit, constraints = c(2,4), vcov = mfor_CR2)
wald_test(mfor_fit, constraints = 2:5, vcov = mfor_CR2)

---

\code{vcovCR.rma.uni}  
\textit{Cluster-robust variance-covariance matrix for a \code{rma.uni} object.}

Description

\code{vcovCR} returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from a \code{rma.uni} object.

Usage

## S3 method for class \code{rma.uni}'
vcovCR(obj, cluster, type, target, inverse_var, 
form = "sandwich", ...)

Arguments

\code{obj}  
Fitted model for which to calculate the variance-covariance matrix

\code{cluster}  
Expression or vector indicating which observations belong to the same cluster. Required for \code{rma.uni} objects.

\code{type}  
Character string specifying which small-sample adjustment should be used, with available options \code{"CR0"}, \code{"CR1"}, \code{"CR1p"}, \code{"CR1S"}, \code{"CR2"}, or \code{"CR3"}. See "Details" section of \code{vcovCR} for further information.
vcovCR.robu

Description

vcovCR returns a sandwich estimate of the variance-covariance matrix of a set of regression coefficient estimates from a \texttt{robu} object.

Usage

```r
## S3 method for class 'robu'
vcovCR(obj, cluster, type, target, inverse_var,
       form = "sandwich", ...)
```
Arguments

- **obj**: Fitted model for which to calculate the variance-covariance matrix.
- **cluster**: Optional expression or vector indicating which observations belong to the same cluster. If not specified, will be set to the studynum used in fitting the `robu` object.
- **type**: Character string specifying which small-sample adjustment should be used, with available options "CR0", "CR1", "CR1p", "CR1s", "CR2", or "CR3". See "Details" section of `vcovCR` for further information.
- **target**: Optional matrix or vector describing the working variance-covariance model used to calculate the CR2 and CR4 adjustment matrices. If not specified, the target is taken to be the inverse of the estimated weights used in fitting the `robu` object.
- **inverse_var**: Optional logical indicating whether the weights used in fitting the model are inverse-variance. If not specified, `vcovCR` will attempt to infer a value.
- **form**: Controls the form of the returned matrix. The default "sandwich" will return the sandwich variance-covariance matrix. Alternately, setting `form = "meat"` will return only the meat of the sandwich and setting `form = B`, where `B` is a matrix of appropriate dimension, will return the sandwich variance-covariance matrix calculated using `B` as the bread.

Value

An object of class `c("vcovCR","clubSandwich")`, which consists of a matrix of the estimated variance of and covariances between the regression coefficient estimates.

See Also

- `vcovCR`

Examples

```r
library(robumeta)
data(hierdat)

robu_fit <- robu(effectsize ~ binge + followup + sreport + age,
                  data = hierdat, studynum = studyid,
                  var.eff.size = var, modelweights = "HIER")
robu_fit

robu_CR2 <- vcovCR(robu_fit, type = "CR2")
robu_CR2
coef_test(robu_fit, vcov = robu_CR2, test = c("Satterthwaite", "saddlepoint"))

Wald_test(robu_fit, constraints = c(2,4), vcov = robu_CR2)
Wald_test(robu_fit, constraints = 2:5, vcov = robu_CR2)
```
Wald_test reports Wald-type tests of linear contrasts from a fitted linear regression model, using a sandwich estimator for the variance-covariance matrix and a small sample correction for the p-value. Several different small-sample corrections are available.

Usage

Wald_test(obj, constraints, vcov, test = "HTZ", ...)

Arguments

obj Fitted model for which to calculate Wald tests.
constraints List of one or more constraints to test. See details below.
vcov Variance covariance matrix estimated using vcovCR or a character string speci-
fying which small-sample adjustment should be used to calculate the variance-
covariance.
test Character vector specifying which small-sample correction(s) to calculate. The following corrections are available: "chi-sq", "Naive-F", "HTA", "HTB", "HTZ", "EDF", "EDT". Default is "HTZ".
... Further arguments passed to vcovCR, which are only needed if vcov is a charac-
ter string.

Details

Constraints can be specified as character vectors, integer vectors, logical vectors, or matrices.

Value

A list of test results.

See Also

vcovCR
Index

*Topic datasets
  AchievementAwardsRCT, 2
  dropoutPrevention, 6
  MortalityRates, 8
  SATcoaching, 9

AchievementAwardsRCT, 2
coef_test, 4
conf_int, 5
dropoutPrevention, 6
glm, 10, 12
gls, 10, 13
impute_covariance_matrix, 7
ivreg, 14
lm, 10, 16
lme, 10, 17
MortalityRates, 8
plm, 10, 19
rma.mv, 10, 21
rma.uni, 10, 22
robu, 10, 23, 24
SATcoaching, 9
vcovCR, 4–6, 10, 10, 13–25
vcovCR.glm, 12, 12
vcovCR.gls, 12, 13
vcovCR.ivreg, 14
vcovCR.lm, 12, 16
vcovCR.lme, 12, 17
vcovCR.mlm, 18
vcovCR.plm, 12, 19
vcovCR.rma.mv, 12, 21
vcovCR.rma.uni, 12, 22
Wald_test, 25
vcovCR.robu, 12, 23
vcovVC, 10
vcovHC, 10
vcovCR.Nglm, 12, 12
vcovCR.Ngls, 12, 13
vcovCR.Nivreg, 14
vcovCR.Nlm, 12, 16
vcovCR.Nlme, 12, 17
vcovCR.Nmlm, 18
vcovCR.Nplm, 12, 19
vcovCRNrmaNmvm, 12, 21
vcovCRNrmaNuni, 12, 22
vcovCRNrmaNuni, 12, 22