

Package ‘fereg’

October 22, 2026

Title Fast Fixed-Effects Estimation with the 'fixest' Interface

Version 0.4.0

Description Ordinary least squares, two-stage least squares, Poisson and logit regressions with any number of high-dimensional fixed effects, interactions of fixed effects, varying slopes, weights and iid, heteroskedasticity-robust (HC1, HC2, HC3), multiway-clustered, CRV3, Newey-West, Driscoll-Kraay or Conley standard errors. The formula syntax, arguments, results and methods follow the 'fixest' package and reproduce its estimates. The computations run in a parallel 'Rust' library that eliminates the largest fixed effect exactly and solves for the others by preconditioned conjugate gradient.

License MIT + file LICENSE

URL <https://github.com/leoliu0/fereg>

BugReports <https://github.com/leoliu0/fereg/issues>

Depends R (>= 4.2)

Imports stats

Suggests fixest, testthat (>= 3.0.0), data.table, generics, Matrix

SystemRequirements Cargo (Rust's package manager), rustc (>= 1.85)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.1.1

NeedsCompilation yes

Author Leo Liu [aut, cre, cph],
Authors of the bundled Rust crates [ctb, cph] (see inst/AUTHORS for
each crate's authors and license)

Maintainer Leo Liu <auleo.liu@gmail.com>

Repository CRAN

Date/Publication 2026-10-22 16:30:02 UTC

Contents

etable	2
feols	3
fepois	5
fereg-methods	8
fereg_multi	10
fitstat	11
fixef	12
r2	12
se	13
ssc	14
trade_toy	15
vcov_requests	15
Index	18

etable	<i>Table of estimation results</i>
--------	------------------------------------

Description

Side-by-side table of several estimations in `fixest`’s `etable()` layout: coefficients with standard errors in parentheses and significance stars, fixed effects, variance type and fit statistics.

Usage

```
etable(  
  ...,  
  vcov = NULL,  
  ssc = NULL,  
  tex = FALSE,  
  digits = 4,  
  digits.stats = 5,  
  fitstat = NULL,  
  keep = NULL,  
  drop = NULL,  
  family = NULL  
)
```

Arguments

- ... fereg or fereg_multi objects (named arguments set the column titles).
- vcov, ssc Variance for every model (see [summary\(\)](#)); vcov may also be a list with one variance per model, as in `fixest`.
- tex Return LaTeX code instead of a data frame.
- digits Significant digits of coefficients and standard errors.

digits.stats	Digits of the fit statistics, as in fixest: a number (significant digits, at least one decimal for large values), or "rN" (round to N decimals) / "sN".
fitstat	Fit statistics (see <code>fitstat()</code>). The default is fixest's: observations, R2 and within R2 (adjusted R2 without fixed effects) when every model is OLS, else observations, squared correlation, pseudo R2 and BIC.
keep, drop	Character vectors of regular expressions (Perl syntax) selecting the coefficient rows, as in fixest: keep shows the coefficients matching any of them, then drop removes those matching any of them. Names are matched as displayed (the intercept is "Constant"); a leading "%" matches the original names instead ("(Intercept)"), and a leading "!" negates the expression.
family	Show the row "Family" (OLS, Poisson, Logit); by default only when the models are of different families.

Value

A data frame (class `fereg_etable`) or, with `tex = TRUE`, a character string (class `fereg_etable_tex`).

Examples

```
data(trade_toy, package = "fereg")
m1 <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
m2 <- fepois(count ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
etable(m1, m2, vcov = ~id)
etable(m1, tex = TRUE)
```

feols

Fixed-effects OLS and IV estimation

Description

Drop-in replacement for `fixest::feols()`: same formula syntax, arguments and defaults, same numbers. Estimation runs in the `fereg` Rust core.

Usage

```
feols(
  fml,
  data,
  vcov,
  weights,
  offset,
  subset,
  split,
  fsplit,
  cluster,
  se,
  ssc,
```

```

panel.id,
fixef.rm = "perfect_fit",
fixef.tol = 1e-08,
fixef.iter = 16000,
collin.tol = 1e-09,
nthreads = getOption("fereg.nthreads", default_threads()),
lean = FALSE,
notes = getOption("fereg.notes", TRUE),
...
)

```

Arguments

<code>fml</code>	A formula $y \sim x_1 + x_2 \mid fe_1 + fe_2 \mid endo \sim inst$. The fixed-effects part accepts interactions $fe_1^{fe_2}$ and varying slopes $fe[x]$ (fixed effect and slope) or $fe[[x]]$ (slope only). The right-hand side accepts <code>i()</code> , <code>factor()</code> , <code>I()</code> , functions of variables, interactions and <code>-1</code> . Several dependent variables $c(y_1, y_2)$ and the stepwise functions <code>sw()</code> , <code>sw0()</code> , <code>csw()</code> , <code>csw0()</code> (right-hand side and fixed-effects part) give multiple estimations.
<code>data</code>	A data frame (or a list of equal-length vectors).
<code>vcov</code>	Variance type, as in <code>fixest</code> : a string (" <code>iid</code> ", the default; " <code>hetero</code> " (HC1), " <code>hc2</code> ", " <code>hc3</code> ", " <code>cluster</code> " (first <code>panel.id</code> variable, else first fixed effect), " <code>twoway</code> ", " <code>threeway</code> ", " <code>fourway</code> ", " <code>NW</code> " / " <code>newey_west</code> ", " <code>DK</code> " / " <code>driscoll_kraay</code> ", " <code>conley</code> ", and <code>fereg</code> 's extension " <code>crv3</code> "; case and unique abbreviations are accepted), a one-sided formula ($\sim cl, \sim cl_1 + cl_2$, up to four dimensions, $\sim cl_1^{cl_2}$ clusters on the interaction), a formula <code>type ~ variables</code> (<code>NW ~ id + period</code> , <code>NW(2) ~ id + period</code> , <code>NW ~ period</code> (time series without <code>panel.id</code>), <code>DK(2) ~ period</code> , <code>conley(100) ~ lat + lon</code> , <code>conley("60mi") ~ lat + lon</code> , <code>crv3 ~ g1 + g2; + ssc(...)</code> among the variables sets the small-sample correction) or a request object (see vcov_requests). Variables of Newey-West and Driscoll-Kraay variances default to <code>panel.id</code> , Conley coordinates are guessed from the column names. Rows with missing values in a variance variable are removed (counted as " <code>vcov</code> " in the NA note).
<code>weights</code>	Observation weights: a one-sided formula ($\sim w$), a column name or a vector.
<code>offset</code>	Offset on the link scale: a one-sided formula, a column name or a vector.
<code>subset</code>	Rows to use: logical or integer vector, or a one-sided formula ($\sim year > 2000$).
<code>split, fsplit</code>	A one-sided formula (or column name): one estimation per value of the variable; <code>fsplit</code> adds the full sample first.
<code>cluster, se</code>	Former <code>fixest</code> ways of requesting a variance type (<code>cluster = ~g</code> , <code>se = "hetero"</code>).
<code>ssc</code>	Small-sample correction, see ssc() .
<code>panel.id</code>	Panel identifiers, as in <code>fixest</code> : a formula $\sim unit + time$ or a string " <code>unit, time</code> ". They are the default unit and time of Newey-West / Driscoll-Kraay variances and the default cluster of <code>vcov = "cluster"</code> . (<code>fereg</code> has no lag operators.)
<code>fixef.rm</code>	" <code>perfect_fit</code> " (default; singletons, plus fixed-effect groups with a perfectly fitted outcome in GLMs), " <code>singleton</code> " or " <code>none</code> ".

<code>fixef.tol</code> , <code>fixef.iter</code>	Tolerance and iteration cap of the fixed-effects solver.
<code>collin.tol</code>	Collinearity tolerance (fixest rule).
<code>nthreads</code>	Number of threads; 0 = all logical cores. Defaults to option <code>fereg.nthreads</code> , else all cores, capped at <code>OMP_THREAD_LIMIT</code> when set (2 under R CMD check, i.e. when <code>_R_CHECK_LIMIT_CORES_</code> is set).
<code>lean</code>	If TRUE, residuals, fitted values and the estimation state are not kept (post-hoc variances then need a new estimation).
<code>notes</code>	Print notes about removed observations (option <code>fereg.notes</code> , default TRUE).
<code>...</code>	Not used; unknown arguments are an error.

Value

An object of class `fereg` (or `fereg_multi` for multiple estimations).

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
m
summary(m, vcov = ~id)
```

fepois	<i>Fixed-effects Poisson and logit estimation</i>
--------	---

Description

Drop-in replacements for `fixest::fepois()` and `fixest::feglm()` (Poisson and logit families).

Usage

```
fepois(
  fml,
  data,
  vcov,
  offset,
  weights,
  subset,
  split,
  fsplit,
  cluster,
  se,
  ssc,
  panel.id,
  fixef.rm = "perfect_fit",
  fixef.tol = 1e-08,
```

```

    fixef.iter = 16000,
    glm.iter = 25,
    glm.tol = 1e-08,
    collin.tol = 1e-09,
    nthreads = getOption("fereg.nthreads", default_threads()),
    lean = FALSE,
    notes = getOption("fereg.notes", TRUE),
    ...
)

feglm(
  fml,
  data,
  family = "logit",
  vcov,
  offset,
  weights,
  subset,
  split,
  fsplit,
  cluster,
  se,
  ssc,
  panel.id,
  fixef.rm = "perfect_fit",
  fixef.tol = 1e-08,
  fixef.iter = 16000,
  glm.iter = 25,
  glm.tol = 1e-08,
  collin.tol = 1e-09,
  nthreads = getOption("fereg.nthreads", default_threads()),
  lean = FALSE,
  notes = getOption("fereg.notes", TRUE),
  ...
)

```

Arguments

<code>fml</code>	A formula $y \sim x_1 + x_2 \mid fe_1 + fe_2 \mid endo \sim inst$. The fixed-effects part accepts interactions $fe_1^{fe_2}$ and varying slopes $fe[x]$ (fixed effect and slope) or $fe[[x]]$ (slope only). The right-hand side accepts <code>i()</code> , <code>factor()</code> , <code>I()</code> , functions of variables, interactions and <code>-1</code> . Several dependent variables <code>c(y1, y2)</code> and the stepwise functions <code>sw()</code> , <code>sw0()</code> , <code>csw()</code> , <code>csw0()</code> (right-hand side and fixed-effects part) give multiple estimations.
<code>data</code>	A data frame (or a list of equal-length vectors).
<code>vcov</code>	Variance type, as in <code>fixest</code> : a string (" <code>iid</code> ", the default; " <code>hetero</code> " (HC1), " <code>hc2</code> ", " <code>hc3</code> ", " <code>cluster</code> " (first <code>panel.id</code> variable, else first fixed effect), " <code>twoway</code> ", " <code>threeway</code> ", " <code>fourway</code> ", " <code>NW</code> " / " <code>newey_west</code> ", " <code>DK</code> " / " <code>driscoll_kraay</code> ",

"conley", and fereg's extension "crv3"; case and unique abbreviations are accepted), a one-sided formula ($\sim cl$, $\sim cl1 + cl2$, up to four dimensions, $\sim cl1 \wedge cl2$ clusters on the interaction), a formula type $\sim variables$ ($NW \sim id + period$, $NW(2) \sim id + period$, $NW \sim period$ (time series without `panel.id`), $DK(2) \sim period$, $conley(100) \sim lat + lon$, $conley("60mi") \sim lat + lon$, $crv3 \sim g1 + g2; + ssc(\dots)$ among the variables sets the small-sample correction) or a request object (see [vcov_requests](#)). Variables of Newey-West and Driscoll-Kraay variances default to `panel.id`, Conley coordinates are guessed from the column names. Rows with missing values in a variance variable are removed (counted as "vcov" in the NA note).

<code>offset</code>	Offset on the link scale: a one-sided formula, a column name or a vector.
<code>weights</code>	Observation weights: a one-sided formula ($\sim w$), a column name or a vector.
<code>subset</code>	Rows to use: logical or integer vector, or a one-sided formula ($\sim year > 2000$).
<code>split, fsplit</code>	A one-sided formula (or column name): one estimation per value of the variable; <code>fsplit</code> adds the full sample first.
<code>cluster, se</code>	Former fixest ways of requesting a variance type (<code>cluster = $\sim g$</code> , <code>se = "hetero"</code>).
<code>ssc</code>	Small-sample correction, see ssc() .
<code>panel.id</code>	Panel identifiers, as in <code>fixest</code> : a formula $\sim unit + time$ or a string "unit,time". They are the default unit and time of Newey-West / Driscoll-Kraay variances and the default cluster of <code>vcov = "cluster"</code> . (fereg has no lag operators.)
<code>fixef.rm</code>	"perfect_fit" (default; singletons, plus fixed-effect groups with a perfectly fitted outcome in GLMs), "singleton" or "none".
<code>fixef.tol, fixef.iter</code>	Tolerance and iteration cap of the fixed-effects solver.
<code>glm.tol, glm.iter</code>	IRLS convergence tolerance (relative deviance change) and iteration cap.
<code>collin.tol</code>	Collinearity tolerance (fixest rule).
<code>nthreads</code>	Number of threads; 0 = all logical cores. Defaults to option <code>fereg.nthreads</code> , else all cores, capped at <code>OMP_THREAD_LIMIT</code> when set (2 under R CMD check, i.e. when <code>_R_CHECK_LIMIT_CORES_</code> is set).
<code>lean</code>	If TRUE, residuals, fitted values and the estimation state are not kept (post-hoc variances then need a new estimation).
<code>notes</code>	Print notes about removed observations (option <code>fereg.notes</code> , default TRUE).
<code>...</code>	Not used; unknown arguments are an error.
<code>family</code>	"poisson" / <code>poisson()</code> or "logit" / "binomial" / <code>binomial()</code> .

Value

An object of class `fereg` (or `fereg_multi`).

Examples

```
data(trade_toy, package = "fereg")
p <- fepois(count ~ x1 | id + year, trade_toy, nthreads = 2)
p
```

Description

Accessors, post-hoc variances, predictions and fixed effects of `feols()`, `fepois()` and `feglm()` results, as in `fixest`.

Usage

```
## S3 method for class 'fereg'
summary(
  object,
  vcov = NULL,
  ssc = NULL,
  cluster = NULL,
  se = NULL,
  stage = 2,
  ...
)

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

```
## S3 method for class 'fereg'
vcov(
  object,
  vcov = NULL,
  ssc = NULL,
  attr = FALSE,
  cluster = NULL,
  se = NULL,
  ...
)
```

```
## S3 method for class 'fereg'
confint(
  object,
  parm,
  level = 0.95,
  vcov = NULL,
  ssc = NULL,
  cluster = NULL,
  se = NULL,
  ...
)
```

```

## S3 method for class 'fereg'
nobs(object, ...)

## S3 method for class 'fereg'
residuals(
  object,
  type = c("response", "deviance", "pearson", "working"),
  na.rm = TRUE,
  ...
)

## S3 method for class 'fereg'
fitted(object, type = c("response", "link"), na.rm = TRUE, ...)

## S3 method for class 'fereg'
predict(object, newdata, type = c("response", "link"), ...)

## S3 method for class 'fereg'
logLik(object, ...)

## S3 method for class 'fereg'
AIC(object, ..., k = 2)

## S3 method for class 'fereg'
BIC(object, ...)

## S3 method for class 'fereg'
deviance(object, ...)

## S3 method for class 'fereg'
sigma(object, ...)

```

Arguments

<code>object</code>	A fereg object.
<code>vcov, ssc, cluster, se</code>	Post-hoc variance (same forms as in <code>feols()</code> , including the request objects of <code>vcov_requests</code>); computed from the retained estimation state without re-estimating, except CRV3, whose leave-one-cluster-out refits re-estimate the model on the estimation data (found again from the call). NULL keeps the estimation's variance.
<code>stage</code>	IV only: 1 first stage(s), 2 second stage (default), 1:2 both.
<code>...</code>	Not used.
<code>attr</code>	Add the attributes <code>type</code> , <code>df.K</code> , <code>df.t</code> (and <code>G</code>) to the matrix.
<code>parm</code>	Coefficients (names or indices); all by default.
<code>level</code>	Confidence level.
<code>type</code>	<code>resid()</code> : "response", "deviance", "pearson" or "working"; <code>fitted()</code> / <code>predict()</code> : "response" or "link".

na.rm	FALSE returns one value per row of the estimation data (NA for removed rows).
newdata	A data frame for predictions; fixed-effect levels absent at estimation give NA.
k	AIC penalty per parameter.

Value

summary(): the fereg object with the requested variance (coefficient table in \$coeftable); for IV with stage = 1 or 1:2, the requested stages (a fereg_multi when there are several). coef(): named vector of coefficients. vcov(): variance matrix. confint(): data frame of lower and upper bounds, one row per coefficient. nobs(): number of observations. resid(), fitted(), predict(): numeric vectors. logLik(): object of class logLik. AIC(), BIC(), deviance(), sigma(): numbers.

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
summary(m, vcov = "hetero")
coef(m)
vcov(m, vcov = ~id)
confint(m, level = 0.9)
head(predict(m, newdata = trade_toy))
c(AIC = AIC(m), BIC = BIC(m))
```

fereg_multi	<i>Multiple estimations</i>
-------------	-----------------------------

Description

A fereg_multi object is a list of fereg objects (one per estimation, fixest order: sample, fixed effects, dependent variable, right-hand side). Methods apply to every estimation.

Usage

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

## S3 method for class 'fereg_multi'
x[i]

## S3 method for class 'fereg_multi'
as.list(x, ...)

## S3 method for class 'fereg_multi'
summary(object, ...)

## S3 method for class 'fereg_multi'
```

```
coef(object, ...)

## S3 method for class 'fereg_multi'
nobs(object, ...)
```

Arguments

<code>x, object</code>	A <code>fereg_multi</code> object.
<code>...</code>	Passed to the methods of each estimation.
<code>i</code>	Indices or names of estimations.

Value

`print()`: `x`, invisibly. `[]` and `summary()`: a `fereg_multi` object. `as.list()`: a list of `fereg` objects. `coef()`: data frame with one row per estimation and one column per coefficient. `nobs()`: number of observations of each estimation.

Examples

```
data(trade_toy, package = "fereg")
ms <- feols(y ~ csw(x1, x2) | id + year, trade_toy, nthreads = 2)
ms
coef(ms)
nobs(ms)
summary(ms[2], vcov = ~id)
```

<code>fitstat</code>	<i>Fit statistics</i>
----------------------	-----------------------

Description

Same types and names as `fixest::fitstat()`: `n`, `rmse`, `r2`, `ar2`, `wr2`, `war2`, `ll`, `pr2`, `apr2`, `cor2` (`sq.cor`), `aic`, `bic`, `f`, `my`, `g` and, for IV, `ivf1` (`ivf`), `ivwald1` (`ivwald`), `wh` and `sargan`.

Usage

```
fitstat(x, type, simplify = FALSE, ...)
```

Arguments

<code>x</code>	A <code>fereg</code> object.
<code>type</code>	A character vector of types, or a one-sided formula ($\sim r^2 + ivf1$).
<code>simplify</code>	Return a plain vector when every statistic is a single number.
<code>...</code>	Not used.

Value

An object of class `fereg_fitstat` (a named list), or a vector if `simplify`.

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
fitstat(m, ~ n + r2 + wr2)
```

fixef	<i>Fixed effects of an estimation</i>
-------	---------------------------------------

Description

Fixed-effect coefficients with fixest’s normalization: the first fixed effect is free, and in every connected component one level of each other fixed effect is set to 0 (counted in the references attribute). Varying slopes are listed as fe[[x]].

Usage

```
fixef(object, ...)
```

Arguments

object	A fereg object.
...	Not used.

Value

A list (class fereg.fixef) of named vectors, one per fixed effect / slope.

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
fe <- fixef(m)
summary(fe)
head(fe$year)
```

r2	<i>R-squared variants</i>
----	---------------------------

Description

R-squared variants

Usage

```
r2(x, type = "all")
```

Arguments

`x` A fereg object.

`type` Types among "r2", "ar2", "wr2", "war2", "pr2", "apr2", "cor2", or "all".

Value

A named numeric vector (NA where a type does not apply).

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
r2(m)
```

se	<i>Coefficient table accessors</i>
----	------------------------------------

Description

`se()`, `tstat()`, `pvalue()` and `coeftable()` as in `fixest`.

Usage

```
se(object, ...)

tstat(object, ...)

pvalue(object, ...)

coeftable(object, ...)
```

Arguments

`object` A fereg (or `fereg_multi`) object.

`...` Passed to [summary\(\)](#) (`vcov`, `ssc`, `cluster`).

Value

Named vectors, or the coefficient matrix for `coeftable()`.

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
se(m)
pvalue(m, vcov = "hetero")
coeftable(m, vcov = ~id)
```

SSC

Small-sample correction

Description

Same arguments and defaults as `fixest`'s `ssc()`. As in `fixest`, the correction depends on the variance type: heteroskedasticity-robust, Newey-West, Driscoll-Kraay and Conley variances use `K.fixef = "full"` (no nesting), Newey-West and Driscoll-Kraay apply `G.adj` with `G` = the number of time periods, Conley uses $n / (n - K)$, and HC2 / HC3 use no correction at all.

Usage

```
ssc(
  K.adj = TRUE,
  K.fixef = "nonnested",
  K.exact = FALSE,
  G.adj = TRUE,
  G.df = "min",
  t.df = "min",
  adj,
  fixef.K,
  cluster.adj,
  cluster.df
)
```

Arguments

<code>K.adj</code>	Multiply the variance by $(n - 1) / (n - K)$ (iid, clustered, Newey-West, Driscoll-Kraay) or $n / (n - K)$ (heteroskedasticity-robust, Conley).
<code>K.fixef</code>	How fixed effects count in <code>K</code> : "nonnested" (fixed effects nested in a cluster variable do not count), "full" or "none".
<code>K.exact</code>	Count the exact number of redundant fixed-effect coefficients.
<code>G.adj</code>	Multiply clustered variances by $G / (G - 1)$ (Newey-West and Driscoll-Kraay: $T / (T - 1)$ with T time periods).
<code>G.df</code>	"min" (one factor with the smallest number of clusters) or "conventional" (each cluster dimension uses its own G).
<code>t.df</code>	Degrees of freedom of the t distribution with clustered, Newey-West and Driscoll-Kraay variances: "min" ($G - 1$, or $T - 1$), "conventional" ($n - K$) or a number.
<code>adj</code> , <code>fixef.K</code> , <code>cluster.adj</code> , <code>cluster.df</code>	Former <code>fixest</code> names of <code>K.adj</code> , <code>K.fixef</code> , <code>G.adj</code> and <code>G.df</code> .

Value

An object of class `ssc.type`.

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 + x2 | id + year, trade_toy, nthreads = 2)
summary(m, vcov = ~id, ssc = ssc(K.adj = FALSE, G.adj = FALSE))
```

trade_toy	<i>Simulated panel for examples</i>
-----------	-------------------------------------

Description

A small simulated panel: 600 observations of 60 units over 10 years.

Usage

```
data(trade_toy)
```

Format

A data frame with columns id, year, x1, x2, z (instrument), xe (endogenous regressor), y, count (Poisson outcome) and binary (0/1 outcome).

vcov_requests	<i>Variance requests</i>
---------------	--------------------------

Description

fixest’s request objects and helpers for the vcov argument of [feols\(\)](#), [fepois\(\)](#), [feglm\(\)](#), [summary\(\)](#), [vcov\(\)](#), [confint\(\)](#) and [etable\(\)](#). They return objects of class `fixest_vcov_request` (the same structure as `fixest`’s, so both packages’ requests are accepted). Called with an estimation as first argument, the `vcov_*`() functions return its variance matrix instead, as in `fixest`.

Usage

```
NW(lag = NULL)

newey_west(lag = NULL)

DK(lag = NULL)

driscoll_kraay(lag = NULL)

conley(cutoff = NULL, pixel = NULL, distance = NULL)

vcov_hetero(
  x,
```

```

    type = "hc1",
    exact = TRUE,
    boot.size = NULL,
    ssc = NULL,
    vcov_fix = TRUE
)

vcov_cluster(x, cluster = NULL, ssc = NULL, vcov_fix = TRUE)

vcov_crv3(x, cluster = NULL, ssc = NULL)

vcov_NW(x, unit = NULL, time = NULL, lag = NULL, ssc = NULL, vcov_fix = TRUE)

vcov_DK(x, time = NULL, lag = NULL, ssc = NULL, vcov_fix = TRUE)

vcov_conley(
  x,
  lat = NULL,
  lon = NULL,
  cutoff = NULL,
  pixel = 0,
  distance = "triangular",
  ssc = NULL,
  vcov_fix = TRUE
)

```

Arguments

lag	Number of lags (NULL = the default rule).
cutoff, pixel, distance	Conley options, see Details.
x	An estimation (fereg object), or omitted to build a request. As in fixest, when the first argument is not an estimation it is taken as the next argument (vcov_NW("id", "t")).
type	vcov_hetero(): "hc1", "hc2" or "hc3".
exact, boot.size	vcov_hetero(): fereg always computes the exact leverage (exact = FALSE is not supported; boot.size is not used).
ssc	A small-sample correction (ssc()); overrides the ssc argument of the caller.
vcov_fix	Must be TRUE (fereg always applies fixest's positive-definiteness fix).
cluster	Cluster variables: a character vector or a one-sided formula.
unit, time	Unit and time variables of Newey-West / Driscoll-Kraay (names or one-sided formulas); by default the variables of panel.id.
lat, lon	Latitude and longitude variables (names or one-sided formulas, degrees).

Details

- `NW(lag)` / `newey_west(lag)`: Newey-West (Bartlett kernel). With a unit and a time variable (`vcov = NW ~ id + period`, or `panel.id`), only pairs of observations of the same unit enter; with a time variable only (`NW ~ period` without `panel.id`) it is the time-series estimator. Time is the dense rank of the time variable over the estimation sample (gaps are collapsed). Default lag: `floor(T^(1/4))` for panels, `fixest::bwNeweyWest()` rule for a time series. Label Newey-West (`L=<lag>`).
- `DK(lag)` / `driscoll_kraay(lag)`: Driscoll-Kraay on the per-period score sums; default lag `floor(T^(1/4))`. Label Driscoll-Kraay (`L=<lag>`).
- `conley(cutoff, pixel, distance)`: Conley spatial HAC with `fixest`'s uniform kernel. `cutoff` is in kilometers (a number) or a string such as "100km" / "60mi"; NULL deduces it from the coordinates as `fixest` does. `pixel` (same unit as `cutoff`, default 0) aggregates the scores on a grid first; `distance` is "triangular" (default, `fixest`'s flat-earth distance) or "spherical" (great-circle). Latitude and longitude come from the formula (`conley(100) ~ lat + lon`) or are guessed from the column names (`lat`, `latitude`, `lat_*`; `lon`, `lng`, `long`, `longitude`, `lon_*`, ...). Label Conley (`<cutoff><km|mi>`).
- `vcov_hetero(type)`: "hc1" (heteroskedasticity-robust, default), "hc2" or "hc3". HC2 and HC3 use the exact leverage of the full model (fixed effects included) and no small-sample correction; they are not defined for IV estimations.
- `vcov_cluster(cluster)`: clustering on up to four variables (names or a formula; `a^b` interacts).
- `vcov_crv3(cluster)`: **fereg extension** (`fixest` has no CRV3): leave-one-cluster-out jackknife centered at the full-sample coefficients (`pyfixest`'s CRV3), with `fixest`'s cluster small-sample correction and multiway inclusion-exclusion. Also requested with `vcov = "crv3"` (clusters guessed as for "cluster") or `vcov = crv3 ~ g1 + g2`. CRV3 needs the leave-one-out refits: after estimation (`summary(m, vcov = "crv3")`) the model is re-estimated.

Value

A `fixest_vcov_request` object, or the variance matrix of `x`.

Examples

```
data(trade_toy, package = "fereg")
m <- feols(y ~ x1 | id + year, trade_toy, panel.id = ~id + year, nthreads = 2)
summary(m, vcov = NW(1))
summary(m, vcov = DK ~ year)
summary(m, vcov = vcov_hetero(type = "hc2"))
```

Index

[.fereg_multi (fereg_multi), 10

AIC.fereg (fereg-methods), 8
as.list.fereg_multi (fereg_multi), 10

BIC.fereg (fereg-methods), 8

coef.fereg (fereg-methods), 8
coef.fereg_multi (fereg_multi), 10
coeftable (se), 13
confint.fereg (fereg-methods), 8
conley (vcov_requests), 15

deviance.fereg (fereg-methods), 8
DK (vcov_requests), 15
driscoll_kraay (vcov_requests), 15

etable, 2
etable(), 15

feglm (fepois), 5
feglm(), 8, 15
feols, 3
feols(), 8, 9, 15
fepois, 5
fepois(), 8, 15
fereg-methods, 8
fereg_multi, 10
fitstat, 11
fitstat(), 3
fitted.fereg (fereg-methods), 8
fixef, 12
fixest::feglm(), 5
fixest::feols(), 3
fixest::fepois(), 5

logLik.fereg (fereg-methods), 8

newey_west (vcov_requests), 15
nobs.fereg (fereg-methods), 8
nobs.fereg_multi (fereg_multi), 10

NW (vcov_requests), 15

predict.fereg (fereg-methods), 8
print.fereg_multi (fereg_multi), 10
pvalue (se), 13

r2, 12
residuals.fereg (fereg-methods), 8

se, 13
sigma.fereg (fereg-methods), 8
ssc, 14
ssc(), 4, 7, 16
summary(), 2, 13, 15
summary.fereg (fereg-methods), 8
summary.fereg_multi (fereg_multi), 10

trade_toy, 15
tstat (se), 13

vcov(), 15
vcov.fereg (fereg-methods), 8
vcov_cluster (vcov_requests), 15
vcov_conley (vcov_requests), 15
vcov_crv3 (vcov_requests), 15
vcov_DK (vcov_requests), 15
vcov_hetero (vcov_requests), 15
vcov_NW (vcov_requests), 15
vcov_requests, 4, 7, 9, 15