

Package ‘clusterIV’

October 1, 2026

Title Clustered Instrumental Variables Estimation and Inference

Version 0.2.0

Description Implements instrumental variables estimation and inference for one endogenous regressor and one-way clustered errors. Includes the cluster-jackknife IV estimator (CJIVE) of Frandsen, Leslie and McIntyre (2025) <[doi:10.1162/rest.a.263](https://doi.org/10.1162/rest.a.263)> and the cluster-jackknife Anderson-Rubin and score tests of Ligtenberg (2025) <[doi:10.48550/arXiv.2306.08559](https://doi.org/10.48550/arXiv.2306.08559)>, which are robust to weak and many instruments. Supports multiple excluded instruments, covariates, precision weights, and high-dimensional fixed effects.

License MIT + file LICENSE

Encoding UTF-8

Imports stats

Suggests generics, knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.3

URL <https://github.com/atal-kat/Clustered-Estimation-and-Inference>

BugReports <https://github.com/atal-kat/Clustered-Estimation-and-Inference/issues>

NeedsCompilation no

Author Atal Katawazi [aut, cre]

Maintainer Atal Katawazi <atalkatawazi@gmail.com>

Repository CRAN

Date/Publication 2026-10-01 21:40:09 UTC

Contents

cjar	2
cjive	12
cjscore	18
clusterIV-tidiers	25

iv_compare	27
iv_infer	30
plot.cjar	34

Index	38
--------------	-----------

cjar	<i>Cluster jackknife Anderson-Rubin test (CJAR)</i>
------	---

Description

Tests hypotheses on the structural coefficient of a single endogenous regressor and inverts the test into a confidence set, using the cluster jackknife Anderson-Rubin statistic of Ligtenberg (2025). The test is cluster-robust and remains valid under weak and under many instruments – the regime in which the usual IV t-statistic (including the one reported by [cjive](#)) is unreliable – and the confidence set is obtained by analytic, grid-free test inversion. Boundary candidates come from a quartic rejection equation and the variance polynomial needed to implement the test’s degenerate-variance rule. The inversion’s coefficient identities are algebraic; roots, endpoints and interval membership are evaluated in floating point, and coverage comes from the test’s asymptotic theory, not from the inversion itself.

Usage

```
cjar(y, ...)  
  
## Default S3 method:  
cjar(  
  y,  
  x,  
  z,  
  cluster,  
  controls = NULL,  
  fixed_effects = NULL,  
  weights = NULL,  
  intercept = TRUE,  
  level = 0.95,  
  beta0 = 0,  
  calibration = c("chisq", "normal"),  
  variance = c("plain", "crossfit"),  
  ...  
)  
  
## S3 method for class 'formula'  
cjar(  
  formula,  
  data,  
  cluster,  
  controls = NULL,
```

```

    fixed_effects = NULL,
    weights = NULL,
    subset,
    na.action = stats::na.omit,
    intercept = TRUE,
    level = 0.95,
    beta0 = 0,
    calibration = c("chisq", "normal"),
    variance = c("plain", "crossfit"),
    ...
)

## S3 method for class 'cjar'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

## S3 method for class 'cjar'
confint(object, parm, level = object$level, ...)

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

## S3 method for class 'summary.cjar'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

```

Arguments

y	Outcome (numeric vector). Factor and character outcomes are rejected. The generic dispatches on this argument; a formula as the first argument selects the formula method.
...	Must be empty. Unknown, misspelled and unnamed arguments are errors.
x	Single endogenous regressor (numeric vector). Factor and character regressors are rejected; for the print method, a fitted "cjar" object.
z	Instruments: a numeric vector/matrix, or a factor/character grouping vector for a judge design. A grouping instrument uses reference coding whenever an intercept or any fixed effect is partialled out. It uses one column for every level only when <code>intercept = FALSE</code> and no fixed effects are supplied. Identical to the z argument of cjive .
cluster	Cluster identifiers (length n). In the formula method all four forms work: a bare column name, a one-sided formula (<code>~ g</code>), a column name as a string, or a full vector (see cjive). The test allows arbitrary dependence within clusters; validity rests on independence across them. A cluster formula must contain exactly one bare variable; multiway formulas and arithmetic combinations of cluster codes are not supported.
controls	Optional exogenous covariates: a matrix or data frame, or a one-sided formula in the formula method. Partialled out of y, x and every instrument by Frisch-

	Waugh-Lovell before the test, the package-wide convention. Reach for it when the instruments are valid only conditionally on covariates. See ‘Controls and fixed effects’ below for what the theory does and does not cover.
fixed_effects	Optional high-dimensional fixed effects to absorb: a factor, or a list/data frame of factors (one per dimension). In the formula method a third <code> </code>-separated part (<code>y ~ x z fe1 + fe2</code>) or a one-sided formula (<code>fixed_effects = ~ fe1 + fe2</code>) is also accepted. Absorbed by the guarded, matrix-free rowsum-based joint projection used in cjive; a result is returned only after its numerical refinement and orthogonality checks pass. Cluster fixed effects are the theoretically safest choice here (Ligtenberg 2025, Section 5.3); see ‘Controls and fixed effects’.
weights	Optional finite, strictly positive numeric precision weights; factor and character weights are rejected. <code>y</code>, <code>x</code> and the instruments are transformed by <code>sqrt(weights)</code> before partialling out, so the result is the CJAR test of the transformed (weighted) model, mirroring the cjive convention. In the formula method the same four forms as <code>cluster</code> work, including a bare data-frame column name.
intercept	One non-missing logical value; partial out an intercept (default <code>TRUE</code>). Disable only when the inputs are already residualised.
level	One finite numeric confidence level strictly between 0 and 1 (default 0.95). Under <code>calibration = "chisq"</code> the critical value is negative exactly when <code>level</code> is below $F_{\chi_k^2}(k)$ – which exceeds 0.5 for every k, since the chi-square mean is larger than its median – so levels slightly above 0.5 can still yield a negative critical value. The test remains a valid one-sided test there and the inversion handles it (a warning is issued); conventional levels (0.9 and above) never hit this regime.
beta0	One finite numeric null value of the structural coefficient at which the statistic and p-value are reported (default 0, the no-effect null). The confidence set is always computed regardless of <code>beta0</code>; set this when a specific hypothesised effect is of interest.
calibration	Critical-value and p-value calibration. <code>"chisq"</code> (the default) uses the shifted-and-scaled chi-square value $c_\alpha = (\chi_{k,1-\alpha}^2 - k)/\sqrt{2k}$, the recommendation of Ligtenberg (2025, p. 8); <code>"normal"</code> uses the standard normal quantile $z_{1-\alpha}$. See ‘Critical values’ below for when the two differ.
variance	Variance estimator studentising the statistic and inverted into the confidence set. <code>"plain"</code> (the default) is the feasible estimator of Ligtenberg (2025, eq. 6), the unchanged default path; <code>"crossfit"</code> is the cross-fit estimator of Section 5.4 (eq. 7), which replaces each cluster’s residual with a leave-two-clusters-out fitted proxy, making the variance estimator unbiased at every β under the paper’s assumptions – the plain estimator is inflated away from the true value, which costs power at distant alternatives (the paper’s motivation, crediting Mikusheva and Sun 2022). See ‘Cross-fit variance’ below for the conventions this option carries (no sign guarantee on the variance estimate; conservative acceptance where it is non-positive) and its cost of one leave-out solve per cluster pair.
formula	A model formula in either restricted layout documented in cjive: <code>y ~ x z</code> (optionally <code>y ~ x z fe</code>) or <code>y ~ exog fe endo ~ inst</code>. Formula sections are additive lists of bare variable names, with exactly one outcome and one endogenous regressor; arithmetic, transformed terms and subtraction are rejected except for the <code>0/-1</code> intercept markers.

data	A data frame in which to evaluate the formula.
subset	Optional expression selecting the rows to use, evaluated in data (formula method only), as in lm .
na.action	How to treat missing values (formula method only); see cjive . The default <code>na.omit</code> drops incomplete rows in a single complete-cases filter and records the count as <code>n_dropped</code> ; the vector interface requires clean input.
digits	Number of significant digits to print.
object	A fitted "cjar" object.
parm	Ignored (the set concerns the single structural coefficient).

Details

Validated design: Weighting and dense or fixed-effect residualisation happen before the identifying checks. The transformed data must be finite; y , x and every instrument column must retain usable variation; and the residualised instrument matrix must have full numerical column rank. Scale- and dimension-aware checks report absorbed or collinear instruments rather than silently dropping them. See [cjive](#) for the shared input and restricted-formula contract.

The test: With $\varepsilon(\beta) = y - x\beta$ on the partialled data, the statistic is the centred quadratic form $Q(\beta) = \varepsilon(\beta)' \tilde{P}_Z \varepsilon(\beta)$ studentised by the feasible variance estimator, where $\tilde{P}_Z$ is the instrument projection matrix with its diagonal *cluster blocks* set to zero. Deleting whole blocks – the cluster jackknife – removes every within-cluster error product from the quadratic form; under clustering all of these have non-zero expectation, not just the squares that the ordinary jackknife AR of Mikusheva and Sun (2022) and Crudu, Mellace and Sandor (2021) deletes. The test is *one-sided*: $H_0: \beta = \beta_0$ is rejected when $T(\beta_0)$ exceeds the critical value (Ligtenberg 2025, Theorem 1). Because it studentises a quadratic form rather than a point estimate, no first-stage strength is needed for validity – this is what makes it weak-instrument-robust. There is no two-sided version: large negative values of T do not indicate a violation of the null.

Critical values: The statistic's reference distribution is asymptotic: under the paper's Assumptions and Theorem 1 (Ligtenberg 2025), $T(\beta_0)$ converges to a standard normal as the numbers of instruments and clusters grow. Neither calibration is an exact finite-sample law. The default calibration = "chisq" uses the shifted-and-scaled chi-square value $(\chi_{k,1-\alpha}^2 - k)/\sqrt{2k}$, the paper's suggested approximating calibration for smaller k (p. 8, after Mikusheva and Sun 2022): for small k the statistic is expected to be closer to a shifted-and-scaled chi-square, and the value converges to the normal quantile as k grows. "normal" uses the limit theorem's standard-normal quantile directly. P-values are computed under the matching reference: $1 - F_{\chi_k^2}(k + \sqrt{2k}T)$ or $1 - \Phi(T)$.

Cross-fit variance: `variance = "crossfit"` replaces the plain variance estimator with the cross-fit estimator of Ligtenberg (2025, Section 5.4, eq. 7): inside each cross-cluster product, the residual of a cluster is replaced by $\varepsilon(\beta) - \tilde{\varepsilon}(\beta; g, h)$, where $\tilde{\varepsilon}(\beta; g, h)$ are the fitted values of a regression of $\varepsilon(\beta)$ on the instruments computed *without* clusters g and h . Because the leave-out proxy is independent of both clusters' errors, the estimator is unbiased at every β , not only at the truth – the plain estimator is inflated away from the true value, which costs power at distant alternatives. Everything else is unchanged: statistic numerator, reference distribution, and critical values. The variance remains a quartic polynomial in β ; the confidence set is computed from the

roots of both the rejection-boundary and variance quartics. F_{CJ} is recomputed from the selected variance estimator and therefore can differ between "plain" and "crossfit".

Two conventions come with the option. First, the cross-fit summands are products of two *different* scalars, not squares, so the variance estimate carries no sign guarantee; where $\hat{V}(\beta) \leq 0$ the statistic is undefined and β is *accepted* – the conservative convention of `cjscore`, applied verbatim (reported as `statistic = NA`, `p.value = 1` at β_0 ; in the inversion such regions join the confidence set). Second, the computation solves one leave-out system per cluster pair, on top of the plain $O(nk + Gk^2)$ coefficient work. Each pair solve is dispatched on the stacked size $m = n_g + n_h$ of the two left-out clusters: $O(m^2k + m^3)$ when $m < k$ (a Woodbury identity on the $m \times m$ side) and $O(k^3)$ otherwise, so the $O(G^2)$ pair count – inherent to the estimator – meets a small per-pair cost exactly in the many-instrument, small-cluster designs where it used to bind. Whether the total matters still depends on G , k and the cluster sizes, so benchmark in your own regime. The default remains "plain", whose output is unchanged.

Confidence set shapes: The confidence set collects every β_0 the test does not reject. Its boundary candidates solve low-degree polynomial equations: a quartic rejection boundary together with variance-polynomial roots required by the degenerate and non-positive-variance conventions. The set is computed by analytic, grid-free root isolation plus sign classification. The polynomial coefficients are algebraic identities; the roots and endpoints are floating-point quantities, and interval membership is checked numerically. The set's coverage is the test's asymptotic coverage, not a property of the inversion algebra. For compatibility, shape classifies the tails; `conf_set` and the three explicit topology fields are authoritative. Five labels occur:

"bounded" all endpoints finite: one interval, a union of two disjoint bounded intervals (a legal quartic outcome), possibly with an isolated accepted point (a degenerate row with equal endpoints, where the boundary touches the acceptance region).

"ray" a single unbounded half-line; it arises via two degenerate routes – the family $w_4 = 0$ (which forces $n_2 = 0$, so Q is linear and the two tail signs can differ), and $F_{CJ} = c_\alpha$ exactly (the leading quartic coefficient $n_2^2 - c_\alpha^2 k w_4$ vanishes, the polynomial degenerates to a cubic, and the two tail signs differ) – and carries the same weak-instrument reading as "two_rays".

"two_rays" both tails are accepted. The simplest case is the complement of a bounded interval, $(-\infty, a] \cup [b, \infty)$, but a valid quartic set may also contain a bounded accepted component between the rays.

"whole_line" every β is accepted.

"empty" no β is accepted.

Unbounded sets are not a bug: by the logic of Dufour (1997), any procedure that is valid under weak instruments *must* return an unbounded set with positive probability, precisely when the data cannot distinguish the structural coefficient. An unbounded set is the test telling you the instruments carry too little information at this confidence level.

Instrument strength: F_{CJ} : F_{CJ} is the leading-coefficient, first-stage form of the selected CJAR statistic: it places x in the role of the error and tests instrument irrelevance ($\Pi = 0$). When its selected variance leading coefficient is positive, comparison with the critical value is the algebraic tail-boundedness criterion: in exact arithmetic the test rejects at $\beta = \pm\infty$ if and only if $F_{CJ} > \text{crit}$. Under the plain variance this is the package's primary cluster-jackknife strength diagnostic. Under cross-fit it is recomputed using the cross-fit leading coefficient, can be NA when that coefficient is non-positive, and should be read as a diagnostic for the selected procedure rather than a variance-invariant quantity.

The effective first-stage F (Montiel Olea-Pflueger): Alongside `F_CJ`, every fit reports the clustered *effective F* of Montiel Olea and Pflueger (2013): the first-stage quadratic form $x'P_Zx$ divided by $\text{tr}(\hat{W}_2)$, the trace of the clustered variance estimate of the (orthonormalised) first-stage coefficient vector, with the null of weak instruments rejected when `F_eff` exceeds the simplified-TSLS Patnaik critical value at Nagar-bias tolerance $\tau = 10\%$ and size $\alpha = 5\%$ (at $K_{eff} = 1$ that value is 23.11 – the origin of the familiar “effective F > 23.1” rule of thumb). Two implementation notes. (i) Only the *simplified* TSLS procedure is implemented: the generalized procedure requires $B_e(W, \Omega)$, a supremum of the Nagar-bias ratio needing numerical optimisation; by the paper’s Theorem 1.3 the simplified critical value is conservative ($B_{TSLS}(W, \Omega) \leq 1$), it is what Stata’s `weakivtest` reports as `cSimp`, and it is standard practice. (ii) The variance estimate is the paper’s, with *no* small-sample correction; `weakivtest` multiplies in the usual cluster finite-sample factor $G/(G-1)(n-1)/(n-L)$, so the two normalizations differ by that factor. That factor must be accounted for before comparing statistics. The critical value is unaffected because K_{eff} is scale-invariant in $\hat{W}_2$. If the estimated denominator is exactly zero while the first-stage signal is nonzero, `F_eff` is `Inf`; `K_eff` and its critical value are then undefined and reported as `NA`.

Controls and fixed effects: Three regimes, in decreasing order of theoretical cover (Ligtenberg 2025, Section 5.3). (i) *Cluster-specific controls* – including cluster fixed effects – introduce dependence only within clusters, so the test applies unchanged; this also covers instruments nested inside clusters (e.g. judges within courts). (ii) A *small number of generic controls* partialled out *ex ante* is asymptotically negligible. (iii) The *many-controls* regime is not covered by the CJAR theory: the global partialling-out step uses each observation’s own cluster, and with many controls (easily entered via `fixed_effects`, e.g. thousands of absorbed levels that are not cluster-specific) this reintroduces a bias the cluster jackknife does not touch – the same mechanism Kolesar, Min, Wang and Zhang (2026) document for CJIVE’s plug-in standard errors. The raw nuisance-count ratio is printed as a transparency diagnostic whenever fixed effects are absorbed; no correction is applied. `k_controls` is a raw nuisance-column upper bound: dense control columns plus the explicit intercept when there are no fixed effects, or one intercept plus $L_j - 1$ columns per absorbed dimension. It is not the effective rank-adjusted nuisance dimension: nesting, duplicate dimensions and dense-control collinearity can reduce rank.

Relation to `cjive`: `cjar()` does not center on the CJIVE estimate and does not use its standard error; the confidence set can be asymmetric around the CJIVE point estimate or exclude it entirely under weak identification. The recommended weak-ID-robust workflow reports both: the CJIVE point estimate for magnitude, the CJAR confidence set for inference – `cjive(...)` for the estimate, `cjar(...)` for the set (the two share the partialling conventions and the instrument Gram factorisation, so results refer to the same design). When the CJAR set is bounded and identification is strong, the two typically agree closely; when they disagree, report the pre-specified weak-ID-robust procedure rather than selecting a result after seeing the data. This package uses CJAR as the primary confidence set and CJS as complementary point-null inference.

FAQ:

What does an unbounded confidence set mean, and what should I do? The instruments are too weak for a bounded set at this level (`F_CJ <= crit`, when that diagnostic is defined). Report the set at the pre-specified confidence level; do not replace it after seeing the result with a t-based interval, which can be spuriously short under weak identification.

Why can the set be empty? Only in the bounded regime, when $Q(\beta)$ stays above the acceptance boundary for every β : no value of the structural coefficient is compatible with the data at this

level. This outcome is not, by itself, proof of invalid instruments or misspecification; inspect the maintained assumptions, design, and numerical diagnostics.

Why is the test one-sided? Under the alternative the quadratic form acquires a positive non-centrality, so evidence against the null accumulates only in the right tail; the left tail carries no information about β .

My data aren't clustered – can I still use this? Yes: set `cluster = seq_len(n)`, every cluster a singleton. $\tilde{P}_Z$ is then the projection matrix with a zeroed diagonal and `cjar()` computes the jackknife AR statistic of Mikusheva and Sun (2022, eq. 2) studentised by the plain variance estimator – the Crudu-Mellace-Sandor-type plug-in their paper calls “naive” – with the same one-sided rejection rule (their Section 4.2: the statistic can be negative, and rejection is in the right tail only). Call the result the “Mikusheva-Sun statistic with plain variance”, not the Mikusheva-Sun test proper. Adding `variance = "crossfit"` gives a cross-fit jackknife AR *in the spirit of* Mikusheva and Sun (2022), using Ligtenberg's (2025) leave-two-out construction – still not their test proper: their recommended estimator $\hat{\Phi}_2$ (their eq. 4) reweights leave-nothing-out residual products, Ligtenberg's leaves clusters out, and the two are unbiased by different mechanisms and not algebraically equal; no setting of the package produces $\hat{\Phi}_2$ exactly. The leave-three-out estimator of Anatolyev and Solvsten (2023) is a further member of the same family and is not implemented. (The Crudu-Mellace-Sandor test itself uses a different centring and does not arise as a reduction of `cjar()`.) The print method labels this case “independent data”. The Stata implementation of their cross-fit test is `manyweakiv` (Sun, 2023, SSC).

Does a grouping-factor z work the same way as in cjive? Yes: with an intercept or absorbed fixed effects the factor is reference-coded; only without either does every level get a column. Dense paths assess support from the encoded, residualized instrument matrix. Only the explicit `cjive(..., method = "leaveout_mean")` shortcut requires every group to have positive weight outside each focal cluster.

Where is the first-stage F? Both first-stage strength statistics are reported, and they are built from the same two ingredients: $E[x' P_Z x] = \pi' Z' Z \pi + \sum_g \text{tr}(P_{[g,g]} \Sigma_g)$, where the second, own-cluster term is of order k and is exactly what the cluster jackknife removes ($E[x' \tilde{P}_Z x] = \pi' Z' Z \pi$). The effective F's denominator, $\text{tr}(\hat{W}_2) = \sum_g \text{tr}(P_{[g,g]} \Sigma_g)$ in expectation, estimates that very own-cluster term. So `F_eff` is a *signal-to-bias* ratio – the familiar fixed- k diagnostic (simplified TSLS procedure, $\tau = 10\%$, $\alpha = 5\%$), tending to 1 under no first stage – while `F_CJ` is a *signal-to-noise* ratio whose comparison with c_α is the algebraic boundedness criterion of the CJAR set. One divides by the own-cluster term, the other subtracts it; as k grows with the total signal held fixed, `F_eff` falls mechanically while the plain-variance `F_CJ` need not. Report both; `F_CJ` is the one whose threshold describes the tails of the selected CJAR set. Nothing in the package branches on `F_eff`, and it deliberately appears in `no glance()` output.

Computation: The polynomial coefficients of $Q(\beta)$ (quadratic) and $\hat{V}(\beta)$ (quartic) are computed once from per-cluster instrument sums in $O(nk + Gk^2)$ time and $O(Gk + k^2)$ memory: one C-level rowsum pass per variable, one triangular backsolve against the Cholesky factor of $Z'Z$ (shared with the `cjive` path), and a handful of $k \times k$ trace contractions. No G-fold R loop, no explicit inverse, and no $n \times n$ or $G \times G$ object is ever formed. Point evaluation of $T(\beta)$, the p-value, the full confidence set and `F_CJ` are all $O(1)$ afterwards. The `maxlev` leverage diagnostic runs in the whitened leave-cluster-out kernel shared with `cjive`, at $O(nk^2 + \sum_g \min(n_g, k)^3)$: one triangular whitening solve, then per cluster the eigenvalues of whichever of the $k \times k$ Gram or the $n_g \times n_g$ projection block is smaller (the Woodbury small-cluster branch of that kernel). It

dominates the coefficient kernel when k is large. The statistic and variance are checked against a direct dense implementation with diagonal cluster blocks explicitly removed.

Value

An object of class "cjar": a list with

statistic, **p.value** the studentised CJAR statistic $T(\beta_0)$ and its one-sided p-value under the chosen calibration. In the degenerate case $\hat{V}(\beta_0) = 0$ the statistic is set to 0 by convention: a zero variance estimate forces $Q(\beta_0) = 0$ in exact arithmetic (both are sums over the same cross-cluster scalars), so T is 0/0 and β_0 is accepted at every nonnegative critical value – all conventional levels; in the negative-critical-value regime of low level the same $T = 0$ rule rejects. The confidence-set inversion applies this rule identically, so set membership and the p-value agree in the degenerate case too. Under **variance = "crossfit"** a non-positive variance estimate at β_0 instead reports **statistic = NA** and **p.value = 1**: the statistic is undefined and β_0 is accepted (see ‘Cross-fit variance’).

term, **beta0**, **level**, **calibration**, **variance_estimator**, **crit** the endogenous-regressor label, null value, confidence level, calibration, variance estimator ("plain" or "crossfit"), and the critical value c_α actually used.

conf_set an $m \times 2$ matrix of interval endpoints (0 rows when empty; $-\text{Inf}/\text{Inf}$ allowed in the first/last row; a row with equal endpoints is an isolated accepted point, where the boundary touches the acceptance region from outside).

shape one of "bounded", "ray", "two_rays", "whole_line", "empty". This is a backwards-compatible tail classification, not a complete topology: "two_rays" may also contain bounded middle components.

n_components, **unbounded_left**, **unbounded_right** the complete number-of-components and tail topology read directly from **conf_set**.

bounded logical; TRUE when every endpoint is finite (including the empty set).

F_CJ the cluster jackknife first-stage strength statistic (NA if its variance term is zero); see ‘Instrument strength’.

n, **G**, **k** observations, clusters, and number of instrument columns after partialling.

F_eff, **K_eff**, **F_eff_crit** the Montiel Olea and Pflueger (2013) effective first-stage F statistic (clustered, no small-sample correction), its effective degrees of freedom (possibly fractional), and the simplified-TSLS Patnaik critical value at $\tau = 10\%$, $\alpha = 5\%$. Reported beside **F_CJ** by **print** and **summary**; see ‘Instrument strength’ and the FAQ entry ‘Where is the first-stage F?’. If the estimated denominator is exactly zero with nonzero signal, **F_eff = Inf** and both **K_eff** and **F_eff_crit** are NA.

ng_max the size of the largest cluster (the advisory thresholds are re-derivable from the stored fields).

maxlev $\max_g \|P_{Z,[g,g]}\|_2$, the Assumption A2(ii) leverage diagnostic; values near 1 mean one cluster nearly spans the instrument space. The paper’s A2(ii) bounds the *square* of this spectral norm, $\|P_{Z,[g,g]}\|_2^2 \leq C < 1$; the package reports the norm itself. The two are equivalent (the eigenvalues of a projection block lie in $[0, 1]$), but the reported number is the norm, not its square.

fe_dims, **fe_levels**, **k_controls**, **n_dropped** as in [cjive](#).

`coef_num`, `coef_var` the numerator coefficients (n_0, n_1, n_2) of $Q(\beta) = n_0 - n_1\beta + n_2\beta^2$ and the variance coefficients $(w_0, \dots, w_4)$ of $\hat{V}(\beta)$, on the original beta scale. They make the object self-contained: $T(\beta)$ at any other β is a two-line evaluation, and `confint` re-inverts at a new level without refitting.

`call` the matched call.

Methods (by generic)

- `print(cjar)`: Print the statistic, the confidence set with its shape, and the F_CJ / leverage diagnostics.
- `nobs(cjar)`: Number of observations used in the test.
- `confint(cjar)`: Confidence set for the structural coefficient. `level` defaults to the level the object was fitted at, where the stored set is returned; at any other level the inversion is recomputed from the stored polynomial coefficients (they are level-free), without refitting.
- `summary(cjar)`: Build a summary object: the full diagnostic picture in one block – statistic, confidence set with a plain-language reading of its shape, the instrument-strength block (F_CJ and the Montiel Olea-Pflueger effective F side by side), design diagnostics, and the advisories that apply.

Functions

- `print(summary.cjar)`: Print method for the summary object.

References

Ligtenberg, J. W. (2025). Inference in clustered IV models with many and weak instruments. arXiv:2306.08559v3. Section 3 and Theorem 1 for the statistic; p. 8 for the feasible variance estimator and the chi-square critical value; Section 5.3 for controls and cluster fixed effects; Section 5.4 (eq. 7) for the cross-fit variance estimator behind `variance = "crossfit"`. The cluster jackknife itself originates with the 2023 first version of this paper.

Mikusheva, A. and Sun, L. (2022). Inference with many weak instruments. *Review of Economic Studies*, 89(5), 2663–2686. The independent-data jackknife AR (eq. 2; the singleton-cluster reduction of CJAR), the naive and cross-fit variance estimators (Section 4.1), the one-sided rejection rule (Section 4.2) and the chi-square critical-value recommendation.

Crudu, F., Mellace, G. and Sandor, Z. (2021). Inference in instrumental variable models with heteroskedasticity and many instruments. *Econometric Theory*, 37(2), 281–310. The variance estimator type Mikusheva and Sun call naive; their own test uses a different centring and is not a reduction of `cjar()`.

Anatolyev, S. and Solvsten, M. (2023). Testing many restrictions under heteroskedasticity. *Journal of Econometrics*, 236(1), 105473. The leave-three-out variance estimator, the third member of the jackknife AR variance family (not implemented).

Dufour, J.-M. (1997). Some impossibility theorems in econometrics with applications to structural and dynamic models. *Econometrica*, 65(6), 1365–1387. Why weak-instrument-valid confidence sets must sometimes be unbounded.

Kolesar, M., Min, P., Wang, W. and Zhang, Y. (2026). Cluster-Robust Inference for Quadratic Forms. arXiv:2602.13537. The many-controls caveat.

Montiel Olea, J. L. and Pflueger, C. (2013). A Robust Test for Weak Instruments. *Journal of Business & Economic Statistics*, 31(3), 358–369. The effective F (eq. 16), the simplified TSLS procedure (Definition 2, eqs. 32–33) and its Table 1 critical values.

See Also

`iv_infer` for the recommended one-call workflow (CJIVE estimate + CJAR confidence set + CJS point-null test with shared preprocessing); `cjive` for the point estimate this test is designed to accompany; `cjscore` for the companion score (LM) test with complementary power properties; `iv_compare` for estimator comparison. Two precomputed worked examples on real data: `vignette("queens-workflow", package = "clusterIV")` and `vignette("miami-bail", package = "clusterIV")`.

Examples

```
## A simulated judge (examiner) design: judge dummies as instruments,
## errors dependent within clusters.
set.seed(42)
G <- 30; ng <- 8; n <- G * ng
cl <- rep(seq_len(G), each = ng)          # cluster identifier
judge <- factor(rep(1:6, length.out = n)) # judge identity (grouping z)
u <- rnorm(G)[cl]                        # cluster-level shock
x <- 0.6 * as.numeric(judge) + u + rnorm(n)
y <- 0.5 * x + u + rnorm(n)

## Strong instruments: a bounded confidence set
ar <- cjar(y, x, judge,
           cluster = cl,          # arbitrary dependence within clusters
           beta0 = 0,            # null reported in the header (default)
           level = 0.95,        # confidence level of the set (default)
           calibration = "chisq") # paper's recommended critical value (default)
ar

## Weak instruments: F_CJ falls below the critical value and the set is
## unbounded -- the honest answer, not a failure (see Details).
xw <- 0.02 * as.numeric(judge) + u + rnorm(n)
yw <- 0.5 * xw + u + rnorm(n)
cjar(yw, xw, judge, cluster = cl)

## Recommended weak-ID-robust workflow: CJIVE point estimate + CJAR set
est <- cjive(y, x, judge, cluster = cl) # point estimate and SE
coef(est)
confint(ar)                             # weak-ID-valid confidence set

## Formula interface, including fixed effects after a second bar
dat <- data.frame(y = y, x = x, judge = judge, cl = cl,
                  f = factor(sample(1:12, n, replace = TRUE)))
cjar(y ~ x | judge | f, data = dat, cluster = ~cl)
```

cjive

Cluster-jackknife instrumental variables estimation (CJIVE)

Description

Computes the cluster-jackknife IV estimator of Frandsen, Leslie and McIntyre (2025) for one endogenous regressor and one or more excluded instrument columns, with cluster-robust inference. The first-stage value for each observation is fitted from a regression that leaves out the observation's entire cluster, which removes the many-instrument bias that survives clustering.

Usage

```

cjive(y, ...)

## Default S3 method:
cjive(
  y,
  x,
  z,
  cluster,
  controls = NULL,
  weights = NULL,
  level = 0.95,
  intercept = TRUE,
  method = c("auto", "dense", "leaveout_mean"),
  fixed_effects = NULL,
  inference = c("asymptotic", "t"),
  ...
)

## S3 method for class 'formula'
cjive(
  formula,
  data,
  cluster,
  controls = NULL,
  weights = NULL,
  level = 0.95,
  intercept = TRUE,
  method = c("auto", "dense", "leaveout_mean"),
  fixed_effects = NULL,
  inference = c("asymptotic", "t"),
  subset,
  na.action = stats::na.omit,
  ...
)

```

```
## S3 method for class 'cjive'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

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

## S3 method for class 'summary.cjive'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

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

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

Arguments

<code>y</code>	Outcome (numeric vector). Factor and character outcomes are rejected. The generic dispatches on this argument; a formula as the first argument selects the formula method (see <code>formula</code>).
<code>...</code>	Must be empty. Unknown, misspelled and unnamed arguments are errors.
<code>x</code>	Single endogenous regressor (numeric vector). Factor and character regressors are rejected; for the print methods, a fitted "cjive" object.
<code>z</code>	Instruments: a numeric vector/matrix, or a factor/character grouping vector for a judge design. A grouping instrument uses reference coding whenever an intercept or any fixed effect is partialled out. It uses one column for every level only when <code>intercept = FALSE</code> and no fixed effects are supplied.
<code>cluster</code>	Cluster identifiers (length <code>n</code>). In the formula methods all four forms work: a bare column name (<code>cluster = cl</code>), a one-sided formula (<code>cluster = ~cl</code>), a column name as a string (<code>cluster = "cl"</code>), or a full vector (<code>cluster = df\$cl</code>); a bare name is looked up in data first, then in the caller. A cluster formula must contain exactly one bare variable; multiway cluster formulas and arithmetic combinations of cluster codes are not supported.
<code>controls</code>	Optional covariates (FLM's X): a matrix or data frame, or a one-sided formula in the formula methods. May be rank deficient (fixed effects are allowed). An intercept is added unless <code>intercept = FALSE</code> .
<code>weights</code>	Optional numeric precision weights. Values must be finite and strictly positive; factor and character weights are rejected. In the formula methods the same four forms as <code>cluster</code> work, including a bare data-frame column name.
<code>level</code>	Confidence level for the reported interval: one finite numeric value strictly between 0 and 1.

intercept	One non-missing logical value; partial out an intercept (default TRUE).
method	One of "auto", "dense", "leaveout_mean". "auto" and "dense" both use the dense Frisch-Waugh-Lovell block-jackknife and are the default. "leaveout_mean" evaluates FLM's printed leave-cluster-out group-mean form and is available only for a grouping-factor z with intercept-only controls; it differs from the default by an intercept term of order n_g/n (about $1/G$ in balanced designs) and is never selected automatically.
fixed_effects	Optional high-dimensional fixed effects to absorb: a factor, or a list/data frame of factors (one per dimension). In the formula methods they can also enter through either formula layout or as a one-sided formula (<code>fixed_effects = ~ fe1 + fe2</code>). Absorbed by a matrix-free, rowsum-based joint projection before the dense partialling out; numerically identical (to tolerance) to entering the corresponding dummies via controls, but without forming any dummy matrix. Absorption spans the intercept within groups, and a result is returned only after normalized refinement and per-dimension weighted orthogonality checks pass.
inference	Critical values and p-values: "asymptotic" (standard normal, the default) or "t" (Student t with $G - 1$ degrees of freedom, a small-cluster reference convention in common use; it is not an exact finite-sample t law under arbitrary within-cluster dependence). The coefficient and standard error are identical under both; only the interval and p-value change.
formula	A model formula in either of two restricted layouts. The legacy layout is $y \sim x \mid z$ (endogenous regressor, then instruments), optionally $y \sim x \mid z \mid fe$ with fixed effects after a second bar; controls enter via the controls argument. The controls-inside- formula layout is $y \sim exog \mid endo \sim inst$ or $y \sim exog \mid fe \mid endo \sim inst$: exogenous controls first, then (optionally) fixed effects, then the IV part $endo \sim inst$, which must come last. This is not a full model-formula grammar: every section accepts only bare variable names joined by +; transformed terms, arithmetic and subtraction are rejected, apart from 0 or -1 as intercept-removal markers in the endogenous/exogenous section. There must be exactly one outcome and one endogenous regressor. Removing the intercept also switches a grouping instrument to full-level coding unless fixed effects are absorbed (their span already contains the intercept). Supplying controls or fixed effects both in the formula and via <code>controls = / fixed_effects =</code> is an error.
data	A data frame in which to evaluate the formula.
subset	Optional expression selecting the rows to use, evaluated in data (formula methods only), as in lm .
na.action	How to treat missing values (formula methods only): na.omit (the default) drops every row with a missing value in any model component – outcome, endogenous regressor, instruments, controls, fixed effects, cluster or weights – in a single complete-cases filter, and records the count as <code>n_dropped</code> (reported by <code>print</code> when non-zero); <code>na.fail</code> stops on any missing value; <code>na.pass</code> leaves them in, so the input validation reports them. The default methods (vector interface) do not handle missing values at all: they require clean input and stop otherwise.
digits	Number of significant digits to print.
object	A fitted "cjive" object.
parm	Ignored (a single coefficient is estimated).

Details

The estimator is the covariance ratio $\hat{\delta} = \widehat{Cov}(Y, \hat{p}) / \widehat{Cov}(D, \hat{p})$ with the cluster-jackknife constructed instrument $\hat{p}$. Covariates are handled by Frisch-Waugh-Lovell: Y , D and each instrument are residualised on the covariates (with an intercept) once, up front, then the estimator runs on the residuals. This dense route is the single convention everywhere, so `cjive()` and `iv_compare` return the identical CJIVE for any design. The weighted, residualised inputs must be finite. After dense FWL or fixed-effect absorption, y , x and every instrument column must retain usable variation, and the residualised instrument matrix must have full numerical column rank. These checks are scale- and dimension-aware; absorbed or collinear columns are reported rather than silently dropped. Point estimation also stops before division when its IV denominator is numerically zero. The leave-cluster-out fits are computed in whitened coordinates ($\tilde{Z} = ZR^{-1}$, with R the Cholesky factor of $Z'Z$, obtained by one triangular solve): per cluster, a $k \times k$ solve when $n_g > k$, while the Woodbury block update on the $n_g \times n_g$ projection block survives as the small-cluster branch ($n_g \leq k$, including singleton clusters) – it is not discarded. The total cost is $O(nk^2 + \sum_g \min(n_g, k)^3)$ time and one $n \times k$ whitened copy of the instruments; the `maxlev` diagnostic is a free by-product of the same per-cluster pass, since the $k \times k$ and $n_g \times n_g$ blocks share their non-zero eigenvalues. The fits agree numerically with the dense leave-cluster-out definition, collapsing when every cluster is a singleton to the observation-level improved JIVE (IJIVE) of Akerberg and Devereux (2009) – the leave-one-out first-stage fit on the covariate-partialled instruments, not the original JIVE of Angrist, Imbens and Krueger (1999), which would jackknife the covariates alongside the instruments. The cluster jackknife originates in the 2023 first version of Ligtenberg (2025); Frandsen, Leslie and McIntyre (2025) use it to construct CJIVE. For a pure judge design, Ligtenberg and Woutersen (2024, Section 2 and Appendix B) show the connection, up to weighting, between CJIVE on judge dummies and 2SLS with a leave-cluster-out mean instrument. The explicit method = "leaveout_mean" path exposes that group-mean form; the default remains the package-wide dense FWL convention described above.

Formula layouts. Both supported layouts describe the same designs. The legacy layout separates the parts with bars only ($y \sim x \mid z \mid fe$, controls via `controls =`); the second layout ($y \sim exog \mid fe \mid endo \sim inst$) carries controls inside the formula. Both use the restricted bare-name, additive syntax described under `formula`; they do not implement a general R or `feols` formula language. Missing values are handled only on the `formula` path (see `na.action`); the vector interface requires clean input.

Fixed effects passed via `fixed_effects` are absorbed by a matrix-free joint projection. Each factor projector uses C-level rowsum group sums. With multiple dimensions, a conjugate-gradient solve applies self-adjoint combinations of those projectors. The result is returned only after freshly normalized refinement finds no material projection correction, every dimension's weighted group score and correction are below tolerance, and an additional complete sweep is stable. A single dimension is one exact projection. If the weighted FE incidence geometry or an observed Krylov direction is too ill-conditioned to support the requested numerical tolerance, absorption stops with a conditioning diagnosis rather than returning the backward certificate as a forward-accuracy claim. The result is numerically identical, to the stated solver tolerance, to entering the corresponding dummy variables through `controls`, but no dummy matrix is ever formed.

A dense control whose post-absorption norm is at most the fixed-effect numerical tolerance relative to its original weighted norm is treated as absorbed before the remaining FWL regression. Surviving control columns are normalized before pivoted QR, so numerical-rank decisions do not depend on their units. Joint collinearity among surviving controls is allowed.

Many controls and the reported standard errors. Kolesar, Min, Wang and Zhang (2026) show

that CJIVE with the plug-in cluster-robust sandwich – exactly what this package reports – can over-reject when the instrument and control dimensions are large relative to n . In one of their Table 1 designs with $n = 600$, rejection at a nominal 5 percent is 7.9 percent with 50 instruments and 50 controls and 53.4 percent with 150 of each; they also report over-rejection under heterogeneous treatment effects. The mechanism is the global partialling-out step, which uses each observation's own cluster, so many controls reintroduce a bias the first-stage cluster jackknife does not touch. Since `fixed_effects` makes it easy to absorb thousands of levels – that is, to enter exactly this regime – the ratio `k_controls/n` is reported by `print.cjive` as a transparency diagnostic. No variance correction is applied.

Value

An object of class "cjive": a list with `coefficient`, `se`, `statistic`, `p.value`, `conf.low`, `conf.high`, `term` (the endogenous-regressor label), `level`, `inference`, the diagnostics `n`, `G`, `k` (number of instrument columns after expansion) and `p` (the retained v0.1.0 compatibility alias for `k`), `path` ("dense" or "leaveout_mean"), `maxlev` (the maximum within-cluster leverage $\max_g \lambda_{\max}(H_g)$), a conditioning diagnostic; NA on the mean path), `F_eff`, `K_eff` and `F_eff_crit` (the clustered Montiel Olea-Pflueger effective first-stage F, its possibly fractional effective degrees of freedom, and the simplified-TSLS critical value at $\tau = 10\%$, $\alpha = 5\%$; if the estimated denominator is exactly zero with nonzero signal, `F_eff` = Inf while `K_eff` and `F_eff_crit` are NA; all three are NA on the "leaveout_mean" path; the details and the 'Where is the first-stage F?' FAQ live in [cjar](#)), `ng_max` (largest cluster size), `fe_dims` and `fe_levels` (number of absorbed fixed-effect dimensions and their total level count; 0 when none), `k_controls` (a raw nuisance-column upper bound: dense control columns plus the explicit intercept when there are no fixed effects, or one intercept plus $L_j - 1$ columns per absorbed dimension; not the effective rank-adjusted nuisance dimension, because nesting, duplicate dimensions and dense-control collinearity can reduce rank), `n_dropped` (rows removed by `na.action`; 0 on the vector interface), and the `call`.

Methods (by generic)

- `print(cjive)`: Print a concise summary of the fit.
- `nobs(cjive)`: Number of observations used in the fit.
- `summary(cjive)`: Build a summary object.
- `coef(cjive)`: Extract the point estimate.
- `vcov(cjive)`: Extract the cluster-robust (co)variance.
- `confint(cjive)`: Confidence interval for the coefficient. `level` defaults to the level the object was fitted at.

Functions

- `print(summary.cjive)`: Print method for the summary object, including the instrument-strength block (the Montiel Olea-Pflueger effective F; `F_CJ` requires a [cjar](#) fit – `iv_infer` reports both in one call).

References

Akerberg, D. A. and Devereux, P. J. (2009). Improved JIVE estimators for overidentified linear models with and without heteroskedasticity. *Review of Economics and Statistics*, 91(2), 351–362. The improved JIVE that the singleton-cluster limit of CJIVE reduces to.

- Angrist, J. D., Imbens, G. W. and Krueger, A. B. (1999). Jackknife instrumental variables estimation. *Journal of Applied Econometrics*, 14(1), 57–67. The original JIVE.
- Frandsen, B., Leslie, E. and McIntyre, S. (2025). Cluster Jackknife Instrumental Variables Estimation. *Review of Economics and Statistics*.
- Ligtenberg, J. W. (2025). Inference in clustered IV models with many and weak instruments. arXiv:2306.08559v3. The 2023 first version introduced the cluster jackknife used by CJIVE.
- Ligtenberg, J. W. and Woutersen, T. (2024). Multidimensional clustering in judge designs. arXiv:2406.09473. Section 2 and Appendix B establish the judge-dummy CJIVE/leave-out-mean connection up to weighting.
- Gaure, S. (2013). OLS with multiple high dimensional category variables. *Computational Statistics and Data Analysis*, 66, 8–18.
- Guimaraes, P. and Portugal, P. (2010). A simple feasible procedure to fit models with high-dimensional fixed effects. *Stata Journal*, 10(4), 628–649.
- Halperin, I. (1962). The product of projection operators. *Acta Scientiarum Mathematicarum (Szeged)*, 23, 96–99.
- Kolesar, M., Min, P., Wang, W. and Zhang, Y. (2026). Cluster-Robust Inference for Quadratic Forms. arXiv:2602.13537.

See Also

[iv_infer](#) for the recommended one-call workflow (CJIVE estimate + CJAR confidence set + CJS point-null test with shared preprocessing); [cjar](#) and [cjscore](#) for the standalone weak-instrument-robust tests; [iv_compare](#) for estimator comparison. Two precomputed worked examples on real data: `vignette("queens-workflow", package = "clusterIV")` and `vignette("miami-bail", package = "clusterIV")`.

Examples

```
set.seed(1)
G <- 40; ng <- 6; n <- G * ng
cl <- rep(seq_len(G), each = ng)
j <- factor(rep(rep(1:4, length.out = ng), G)) # judge identity
u <- rnorm(G)[cl]
x <- as.numeric(j) + u + rnorm(n)
y <- 1.5 * x + u + rnorm(n)
w1 <- rnorm(n); w2 <- rnorm(n)
fit <- cjive(y, x, j, cluster = cl)
print(fit)

## formula interface with controls supplied through the argument
dat <- data.frame(y = y, x = x, j = j, cl = cl, w1 = w1, w2 = w2)
cjive(y ~ x | j, data = dat, cluster = cl, controls = ~ w1 + w2)

## formula interface, controls inside the formula
cjive(y ~ w1 + w2 | x ~ j, data = dat, cluster = cl)
```

cjscore

*Cluster jackknife score test (CJS)***Description**

Tests point hypotheses on the structural coefficient of a single endogenous regressor using the cluster jackknife score (LM) statistic of Ligtenberg (2025, Section 4), with the confidence set obtained by analytic, grid-free test inversion by sign-classifying the quadratic rejection boundary and quadratic variance polynomial; roots and endpoints are floating-point quantities, and coverage is the test's asymptotic coverage. Like [cjar](#) the test is cluster-robust and remains valid under weak and under many instruments. The paper finds complementary power: CJS can perform better near the tested value and under high heteroskedasticity, while CJAR can perform better with weak instruments. For reporting confidence sets, prefer [cjar](#): see 'Score confidence sets and spurious regions' below.

Usage

```
cjscore(y, ...)

## Default S3 method:
cjscore(
  y,
  x,
  z,
  cluster,
  controls = NULL,
  fixed_effects = NULL,
  weights = NULL,
  intercept = TRUE,
  level = 0.95,
  beta0 = 0,
  variance = c("plain", "crossfit"),
  ...
)

## S3 method for class 'formula'
cjscore(
  formula,
  data,
  cluster,
  controls = NULL,
  fixed_effects = NULL,
  weights = NULL,
  subset,
  na.action = stats::na.omit,
  intercept = TRUE,
  level = 0.95,
```

```

    beta0 = 0,
    variance = c("plain", "crossfit"),
    ...
)

## S3 method for class 'cjscore'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

## S3 method for class 'cjscore'
confint(object, parm, level = object$level, ...)

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

## S3 method for class 'summary.cjscore'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

```

Arguments

<code>y</code>	Outcome (numeric vector). Factor and character outcomes are rejected. The generic dispatches on this argument; a formula as the first argument selects the formula method.
<code>...</code>	Must be empty. Unknown, misspelled and unnamed arguments are errors.
<code>x</code>	Single endogenous regressor (numeric vector). Factor and character regressors are rejected; for the print method, a fitted "cjscore" object.
<code>z</code>	Instruments: a numeric vector/matrix, or a factor/character grouping vector for a judge design. A grouping instrument uses reference coding whenever an intercept or any fixed effect is partialled out. It uses one column for every level only when <code>intercept = FALSE</code> and no fixed effects are supplied. Identical to the <code>z</code> argument of cjar and cjive .
<code>cluster</code>	Cluster identifiers (length <code>n</code>). In the formula method all four forms work: a bare column name, a one-sided formula (<code>~ g</code>), a column name as a string, or a full vector (see cjive). The test allows arbitrary dependence within clusters; validity rests on independence across them. A cluster formula must contain exactly one bare variable; multiway formulas and arithmetic combinations of cluster codes are not supported.
<code>controls</code>	Optional exogenous covariates: a matrix or data frame, or a one-sided formula in the formula method. Partialled out of <code>y</code> , <code>x</code> and every instrument by Frisch-Waugh-Lovell before the test, the package-wide convention; the covariate regimes and caveats documented in cjar ('Controls and fixed effects') apply unchanged.
<code>fixed_effects</code>	Optional high-dimensional fixed effects to absorb: a factor, or a list/data frame of factors (one per dimension). In the formula method a third <code> </code> -separated part (<code>y ~ x z fe1 + fe2</code>) or a one-sided formula is also accepted. Absorbed by the same guarded numerical projection as in cjar and cjive .

weights	Optional finite, strictly positive numeric precision weights; factor and character weights are rejected. y , x and the instruments are transformed by <code>sqrt(weights)</code> before partialling out, so the result is the CJS test of the transformed (weighted) model. In the formula method the same four forms as <code>cluster</code> work, including a bare data-frame column name.
intercept	One non-missing logical value; partial out an intercept (default TRUE). Disable only when the inputs are already residualised.
level	One finite numeric confidence level strictly between 0 and 1 (default 0.95), as in cjar .
beta0	One finite numeric null value of the structural coefficient at which the statistic and p-value are reported (default 0, the no-effect null). The confidence set is always computed regardless of <code>beta0</code> . Testing a specific point null is this function's primary use case.
variance	Variance estimator studentising the score. "plain" (the default) is the feasible estimator of Ligtenberg (2025, eq. 6), the unchanged default path; "crossfit" is the cross-fit estimator of Section 5.4 (eq. 7), unbiased under the paper's assumptions at every β rather than only at the truth. The score coefficients are untouched; only the variance polynomial changes, and the conservative non-positive-variance convention below applies unchanged (the cross-fit estimate, like the plain one, carries no sign guarantee). Costs one leave-out solve per cluster pair, dispatched on the stacked size of the two left-out clusters ($O(m^2k+m^3)$ for $m = n_g + n_h < k$, $O(k^3)$ otherwise); see 'Cross-fit variance' in cjar for the shared construction.
formula	A model formula in either restricted layout documented in cjive : $y \sim x \mid z$ (optionally $y \sim x \mid z \mid fe$) or $y \sim exog \mid fe \mid endo \sim inst$. Formula sections are additive lists of bare variable names, with exactly one outcome and one endogenous regressor; arithmetic, transformed terms and subtraction are rejected except for the $0/-1$ intercept markers.
data	A data frame in which to evaluate the formula.
subset	Optional expression selecting the rows to use, evaluated in <code>data</code> (formula method only), as in lm .
na.action	How to treat missing values (formula method only); see cjive . The default <code>na.omit</code> drops incomplete rows in a single complete-cases filter and records the count as <code>n_dropped</code> ; the vector interface requires clean input.
digits	Number of significant digits to print.
object	A fitted "cjscore" object.
parm	Ignored (the set concerns the single structural coefficient).

Details

Validated design: Weighting and dense or fixed-effect residualisation happen before the identifying checks. The transformed data must be finite; y , x and every instrument column must retain usable variation; and the residualised instrument matrix must have full numerical column rank. Scale- and dimension-aware checks report absorbed or collinear instruments rather than silently dropping them. See [cjive](#) for the shared input and restricted-formula contract.

The test: With $\varepsilon(\beta) = y - x\beta$ on the partialled data, the statistic is $S(\beta) = X'\ddot{P}_Z \varepsilon(\beta)/\sqrt{n}$ (Ligtenberg 2025, Section 4), where $\ddot{P}_Z$ is the instrument projection with its diagonal cluster blocks removed – the same cluster jackknife as `cjar`, applied to the score $X'P_Z\varepsilon$ instead of the quadratic form $\varepsilon'P_Z\varepsilon$. It is studentised by the feasible variance estimator of eq. (6) (conditionally unbiased and consistent by the paper’s Theorem 3, under its assumptions) and compared, two-sided, against the asymptotic normal limit given by the paper’s Theorem 2: $H_0: \beta = \beta_0$ is rejected when $LM(\beta_0) > \chi_{1,1-\alpha}^2$. The chi-square(1) reference is asymptotic under the paper’s assumptions, not an exact finite-sample law. The paper’s shifted-and-scaled χ_k^2 critical value is specific to the AR statistic (whose reference distribution depends on k) and is deliberately not carried over; a calibration argument is not offered because the two candidate calibrations – two-sided normal on $S/\sqrt{\hat{V}}$ and χ_1^2 on LM – are the same test.

When to reach for the score test: The score test offers a different power profile while retaining weak-ID robustness (Ligtenberg 2025, Section 4, after Kleibergen 2002). In the paper’s simulations its advantage concentrates near the tested value with a moderate number of strong instruments and under strong heteroskedasticity; under weak instruments the cluster jackknife AR has better power far from the null. Its appendix also finds the cluster jackknife score notably robust to a dominating cluster, where the cluster jackknife AR over-rejects.

Score confidence sets and spurious regions: Inverting a score test can produce confidence sets containing regions far from any plausible parameter value: a score statistic vanishes wherever the underlying objective is stationary, not only near the truth – here, $S(\beta)$ is linear in β and crosses zero at $\beta = s_0/s_1$, a point the test always accepts, however the data were generated (where the variance estimate is positive the statistic is zero there; where it is non-positive the point is accepted by the conservative convention). In addition, the conservative convention can add a closed interval on which the variance estimate is non-positive, possibly disjoint from the main region (see ‘Non-positive variance estimates’). The set reported by `cjscore()` has asymptotic coverage under the paper’s assumptions (no unconditional finite-sample coverage guarantee is claimed), but for interval reporting prefer `cjar`, whose statistic diverges along any fixed alternative and whose plain variance estimator is a sum of squares (its cross-fit variance, like the cross-fit CJS variance, has no pointwise sign guarantee); use `cjscore()` primarily to test pre-specified point nulls, where its complementary power properties are relevant.

Non-positive variance estimates: Under `variance = "plain"`, unlike the plain AR variance (a sum of squares), the score variance estimator’s cross-cluster term $\sum_{g \neq h} c_{gh}c_{hg}$ can be negative, so $\hat{V}^S(\beta) \leq 0$ is possible in finite samples (it requires at least three clusters; the estimator is correct on average and consistent at the true β_0 by the paper’s Theorem 3, so a non-positive value at a plausible β is itself a red flag). Automatic acceptance at such points is the *package*’s conservative convention – the cited papers do not prove or prescribe this rule. The convention, applied identically in the statistic and the inversion, is: β is rejected if and only if $\hat{V}^S(\beta) > 0$ and $S(\beta)^2 > \chi_{1,1-\alpha}^2 \hat{V}^S(\beta)$. Where the variance estimate is positive this is the usual LM test; where it is non-positive the statistic is not defined and the point is accepted – the test declines to reject. For the plain estimator, the leading variance coefficient satisfies $v_2 \geq 0$, so negativity is confined to a bounded beta-window and does not change the tails. The cross-fit variance has no corresponding sign guarantee: a non-positive region can extend into a tail. The same conservative acceptance convention is applied, but the plain-estimator tail statement and boundedness formula below do not carry over without qualification.

Instrument strength: F_CJS: For `variance = "plain"`, in exact arithmetic the confidence set is bounded if and only if $v_2 > 0$ and $F_CJS^2 > \text{crit}$ (both tails rejected), with $F_{CJS} = s_1/\sqrt{v_2}$

– the algebraic leading-coefficient boundedness criterion. Algebraically, F_{CJS}^2 is the CJS statistic applied to the first stage (testing instrument irrelevance, $\Pi = 0$, with x in the role of the outcome)
 – the score analogue of `cjar`’s `F_CJ`, with the same numerator $x' \ddot{P}_Z x$ and the score-form variance in the denominator. Under cross-fit, `F_CJS` is recomputed from the selected variance leading coefficient. It is NA when that coefficient is non-positive; when finite, its comparison with `crit` describes the selected procedure’s two tails rather than a variance-invariant first-stage quantity. Both forms are free by-products of the stored coefficients.

Relation to `cjar` and `cjive`: `cjscore()` shares `cjar`’s entire infrastructure: the partialling conventions, the instrument Gram factorisation, the `maxlev` diagnostic and the advisory thresholds, so results from the two tests and from `cjive` refer to the same design. The recommended workflow remains: `cjive` for the point estimate, `cjar` for the reported confidence set, `cjscore()` for complementary tests of specific point nulls (and as a cross-check: under strong identification all three agree closely).

FAQ:

My data aren’t clustered – can I still use this? Yes: set `cluster = seq_len(n)`, every cluster a singleton. $\ddot{P}_Z$ is then the projection matrix with a zeroed diagonal and `cjscore()` computes the jackknife Lagrange multiplier test of Matsushita and Otsu (2024): statistic, variance estimator and chi-square(1) calibration coincide algebraically (their eqs. 3–4). The print method labels this case “independent data; Matsushita-Otsu 2024”. The identity holds for the default `variance = "plain"` only – `variance = "crossfit"` substitutes Ligtenberg’s (2025, Section 5.4) leave-out variance construction, which is not the Matsushita-Otsu estimator, and the print label says so – and it is an identity for the no-control (or previously partialled) design. Two package conventions sit on top of their test: the conservative acceptance at non-positive variance estimates (‘Non-positive variance estimates’ above), which their paper does not discuss, and the ex-ante global Frisch-Waugh-Lovell partialling of supplied controls, which is not claimed to equal the included-regressor correction analysed by Matsushita and Otsu – with many controls see the caveat in `cjar` (‘Controls and fixed effects’). The validity citation for clustered data remains Ligtenberg (2025, Theorem 2), which nests this test as the singleton case.

Computation: The coefficients of the linear statistic and quadratic variance are computed once from whitened per-cluster instrument sums in $O(nk + Gk^2)$ time and $O(Gk + k^2)$ memory – the same kernel shape as `cjar`, one Gram matrix cheaper. The confidence set is the sign classification of two quadratics (the acceptance polynomial and the variance polynomial); no grid search anywhere. The `maxlev` diagnostic runs in the whitened leave-cluster-out kernel shared with `cjive` and `cjar`, at $O(nk^2 + \sum_g \min(n_g, k)^3)$. The implementation is checked against a direct dense calculation with the diagonal cluster blocks of the instrument projector explicitly removed.

Value

An object of class “`cjscore`”: a list with

`statistic, p.value` the score statistic $LM(\beta_0) = S(\beta_0)^2 / \hat{V}^S(\beta_0)$ and its p-value under the asymptotic χ_1^2 calibration. When $\hat{V}^S(\beta_0) \leq 0$ the statistic is not defined: it is reported as NA and the p-value is 1, so β_0 is accepted by the conservative convention (see ‘Non-positive variance estimates’).

`score, variance` the paper-scaled ingredients $S(\beta_0) = X' \ddot{P}_Z \varepsilon(\beta_0) / \sqrt{n}$ and $\hat{V}^S(\beta_0)$; the variance may be non-positive in finite samples (its cross-cluster term is not a sum of squares).

`term`, `beta0`, `level`, `variance_estimator`, `crit` the endogenous-regressor label, null value, confidence level, variance estimator ("plain" or "crossfit"), and the critical value $\chi^2_{1,1-\alpha} = \text{qchisq}(\text{level}, 1)$.

`conf_set` an $m \times 2$ matrix of interval endpoints (0 rows when empty; $-\text{Inf}/\text{Inf}$ allowed in the first/last row; a row with equal endpoints is an isolated accepted point). Because points where the variance estimate is non-positive are accepted, the set can contain more than one component: the usual LM region plus regions on which $\hat{V}^S \leq 0$ (see ‘Non-positive variance estimates’). Under the plain variance such a region is bounded; under cross-fit it may extend into a tail.

`shape` one of "bounded", "ray", "two_rays", "whole_line", "empty", sharing the [cjar](#) tail-classification vocabulary. It is not a complete topology when bounded middle components coexist with tails.

`n_components`, `unbounded_left`, `unbounded_right` the complete number-of-components and tail topology read directly from `conf_set`.

`bounded` logical; TRUE when every endpoint is finite (including the empty set).

`F_CJS` the score-form first-stage strength statistic (NA if its variance term is zero); see ‘Instrument strength’.

`n`, `G`, `k` observations, clusters, and number of instrument columns after partialling.

`maxlev` $\max_g \|P_{Z,[g,g]}\|_2$, the same cluster-leverage diagnostic reported by [cjar](#).

`F_eff`, `K_eff`, `F_eff_crit`, `ng_max` the Montiel Olea-Pflueger effective first-stage F and its simplified-TSLS critical value, and the largest cluster size, exactly as in [cjar](#) (whose documentation carries the details and the ‘Where is the first-stage F?’ FAQ). If the estimated denominator is exactly zero with nonzero signal, `F_eff` = Inf and both `K_eff` and `F_eff_crit` are NA.

`fe_dims`, `fe_levels`, `k_controls`, `n_dropped` as in [cjive](#).

`coef_score`, `coef_var` the coefficients (s_0, s_1) of $\sqrt{n} S(\beta) = s_0 - s_1\beta$ and (v_0, v_1, v_2) of $n\hat{V}^S(\beta) = v_0 + v_1\beta + v_2\beta^2$, on the original beta scale; they make the object self-contained (`confint` re-inverts at a new level without refitting).

`call` the matched call.

Methods (by generic)

- `print(cjscore)`: Print the statistic, the confidence set with its shape, and the `F_CJS` / leverage diagnostics.
- `nobs(cjscore)`: Number of observations used in the test.
- `confint(cjscore)`: Confidence set for the structural coefficient. `level` defaults to the level the object was fitted at, where the stored set is returned; at any other level the inversion is recomputed from the stored coefficients (they are level-free), without refitting. Mind the spurious-region caveat in `?cjscore`.
- `summary(cjscore)`: Build a summary object: the full diagnostic picture in one block, as in [summary.cjar](#).

Functions

- `print(summary.cjscore)`: Print method for the summary object.

References

Ligtenberg, J. W. (2025). Inference in clustered IV models with many and weak instruments. arXiv:2306.08559v3. Section 4 for the statistic, Assumption 3 and Theorem 2; eq. (6) for the feasible variance estimator and Theorem 3 for its unbiasedness and consistency; Section 5.4 (eq. 7) for the cross-fit variance estimator behind `variance = "crossfit"`; Section 6 and Appendix F for the power and dominant-cluster evidence cited above.

Kleibergen, F. (2002). Pivotal statistics for testing structural parameters in instrumental variables regression. *Econometrica*, 70(5), 1781–1803. The score test’s ancestor for fixed k .

Matsushita, Y. and Otsu, T. (2024). A jackknife Lagrange multiplier test with many weak instruments. *Econometric Theory*, 40(2), 447–470. The independent-data jackknife LM test (eqs. 3–4 for the statistic and its variance estimator), compared against χ^2 critical values – the calibration convention adopted here. At singleton clusters with `variance = "plain"` and no supplied controls, `cjscore()` reduces to this test algebraically (see the FAQ).

Dufour, J.-M. (1997). Some impossibility theorems in econometrics with applications to structural and dynamic models. *Econometrica*, 65(6), 1365–1387. Why weak-instrument-valid confidence sets must sometimes be unbounded.

See Also

`iv_infer` for the recommended one-call workflow (CJIVE estimate + CJAR confidence set + CJS point-null test with shared preprocessing); `cjar` for the companion AR test and the recommended confidence sets; `cjive` for the point estimate; `iv_compare` for estimator comparison. Two precomputed worked examples on real data: `vignette("queens-workflow", package = "clusterIV")` and `vignette("miami-bail", package = "clusterIV")`.

Examples

```
## Simulated judge design, as in ?cjar
set.seed(42)
G <- 30; ng <- 8; n <- G * ng
cl <- rep(seq_len(G), each = ng)      # cluster identifier
judge <- factor(rep(1:6, length.out = n)) # judge identity (grouping z)
u <- rnorm(G)[cl]                    # cluster-level shock
x <- 0.6 * as.numeric(judge) + u + rnorm(n)
y <- 0.5 * x + u + rnorm(n)

## Test a point null with the score test
sc <- cjscore(y, x, judge,
              cluster = cl,      # arbitrary dependence within clusters
              beta0 = 0,        # the point null of interest (default)
              level = 0.95)     # level of the accompanying set (default)
sc

## Recommended workflow: CJIVE estimate, CJAR set, CJS point-null test
est <- cjive(y, x, judge, cluster = cl)
ar <- cjar(y, x, judge, cluster = cl)
coef(est)
confint(ar) # report this set
cjscore(y, x, judge, cluster = cl, beta0 = 1)$p.value # test beta = 1
```

Description

Coherent `tidy()` and `glance()` methods for `cjive`, `cjar`, `cjscore` and `iv_infer` objects. The methods return plain data frames and are registered with **generics** when that optional package is installed. A conventional coefficient table cannot represent every weak-ID-robust confidence-set topology (disjoint or unbounded sets in particular); consumers needing the complete set must read the `conf.set` list-column and the topology columns. No optional package is required at run time, and no adapter is claimed for packages that discard procedure metadata.

Usage

```
tidy.cjive(  
  x,  
  conf.int = TRUE,  
  conf.level = x$level,  
  vcov = NULL,  
  coef_rename = FALSE,  
  ...  
)  
  
glance.cjive(x, gof_map = NULL, ...)  
  
tidy.cjar(  
  x,  
  conf.int = TRUE,  
  conf.level = x$level,  
  vcov = NULL,  
  coef_rename = FALSE,  
  ...  
)  
  
glance.cjar(x, gof_map = NULL, ...)  
  
tidy.cjscore(  
  x,  
  conf.int = TRUE,  
  conf.level = x$level,  
  vcov = NULL,  
  coef_rename = FALSE,  
  ...  
)
```

```
glance.cjscore(x, gof_map = NULL, ...)

tidy.iv_infer(
  x,
  conf.int = TRUE,
  conf.level = x$level,
  vcov = NULL,
  coef_rename = FALSE,
  ...
)

glance.iv_infer(x, gof_map = NULL, ...)
```

Arguments

<code>x</code>	A fitted <code>cjive</code> , <code>cjar</code> , <code>cjscore</code> or <code>iv_infer</code> object.
<code>conf.int</code>	Include scalar <code>conf.low</code> / <code>conf.high</code> endpoints when the set is representable as one finite non-degenerate interval.
<code>conf.level</code>	Confidence level for those endpoints. Test inversion is recomputed from stored coefficients when necessary.
<code>vcov</code> , <code>coef_rename</code>	Compatibility arguments used by downstream table tools. <code>vcov</code> must be <code>NULL</code> ; coefficient renaming is applied after extraction.
<code>...</code>	Must be empty.
<code>gof_map</code>	A downstream goodness-of-fit mapping, accepted and left for the calling tool to apply after extraction.

Details

A row represents exactly one inferential procedure. The `CJIVE` row contains its estimate, cluster-robust standard error, Wald statistic and p-value for the zero null, and Wald interval. Test rows contain no estimate or standard error; their statistic and p-value refer to the explicit `null.value` stored in the same row. `tidy.iv_infer()` returns one row for each component actually present, beginning with `CJIVE`.

A confidence set is flattened into `conf.low`/`conf.high` only when it is one finite, non-degenerate interval. The authoritative endpoint matrix is always retained in the ordinary list-column `conf.set`; the columns `shape`, `n.components`, `unbounded.left`, and `unbounded.right` describe its topology without relying on attributes. In `glance.iv_infer()`, procedure-specific diagnostics use the `cjar.*` and `cjscore.*` prefixes; a CJS strength statistic is therefore never paired with a CJAR critical value or topology. The `variance.estimator` `glance` field is `NA` when no selected jackknife test uses that choice. Unknown or partially matched arguments are errors.

Value

A `data.frame`. `tidy.iv_infer()` has one row per present procedure; the standalone tidiers and all `glance` methods have one row.

See Also

[cjive](#), [cjar](#), [cjscore](#), [iv_infer](#).

iv_compare

Compare IV estimators on a common cluster-robust standard error

Description

Returns OLS, 2SLS, JIVE and CJIVE for the same design, each reported with the same cluster-robust IV sandwich standard-error convention, computed from that row's single constructed instrument; only the constructed instrument differs between rows. The four-row layout follows Table 1 in Frandsen, Leslie and McIntyre (2025); the function does not replicate that table's empirical estimates.

Usage

```
iv_compare(y, ...)

## Default S3 method:
iv_compare(
  y,
  x,
  z,
  cluster,
  controls = NULL,
  weights = NULL,
  level = 0.95,
  intercept = TRUE,
  fixed_effects = NULL,
  inference = c("asymptotic", "t"),
  ...
)

## S3 method for class 'formula'
iv_compare(
  formula,
  data,
  cluster,
  controls = NULL,
  weights = NULL,
  level = 0.95,
  intercept = TRUE,
  fixed_effects = NULL,
  inference = c("asymptotic", "t"),
  subset,
  na.action = stats::na.omit,
  ...
)
```

Arguments

<code>y</code>	Outcome (numeric vector); factor and character outcomes are rejected. A formula as the first argument selects the formula method.
<code>...</code>	Must be empty. Unknown, misspelled and unnamed arguments are errors.
<code>x</code>	Single endogenous regressor (numeric vector); factor and character regressors are rejected.
<code>z</code>	Instruments (numeric matrix/vector or a grouping factor). A factor uses reference coding with an intercept or absorbed fixed effects, and all levels only when neither is present.
<code>cluster</code>	Cluster identifiers (length <code>n</code>). In the formula methods all four forms work: a bare column name (<code>cluster = cl</code>), a one-sided formula (<code>cluster = ~cl</code>), a column name as a string (<code>cluster = "cl"</code>), or a full vector (<code>cluster = df\$cl</code>); a bare name is looked up in data first, then in the caller. A cluster formula must contain exactly one bare variable; multiway cluster formulas and arithmetic combinations of cluster codes are not supported.
<code>controls</code>	Optional covariates (FLM's X): a matrix or data frame, or a one-sided formula in the formula methods. May be rank deficient (fixed effects are allowed). An intercept is added unless <code>intercept = FALSE</code> .
<code>weights</code>	Optional numeric precision weights. Values must be finite and strictly positive; factor and character weights are rejected. In the formula methods the same four forms as <code>cluster</code> work, including a bare data-frame column name.
<code>level</code>	Confidence level for the reported interval: one finite numeric value strictly between 0 and 1.
<code>intercept</code>	One non-missing logical value; partial out an intercept (default <code>TRUE</code>).
<code>fixed_effects</code>	Optional high-dimensional fixed effects to absorb: a factor, or a list/data frame of factors (one per dimension). In the formula methods they can also enter through either formula layout or as a one-sided formula (<code>fixed_effects = ~fe1 + fe2</code>). Absorbed by a matrix-free, rowsum-based joint projection before the dense partialling out; numerically identical (to tolerance) to entering the corresponding dummies via <code>controls</code> , but without forming any dummy matrix. Absorption spans the intercept within groups, and a result is returned only after normalized refinement and per-dimension weighted orthogonality checks pass.
<code>inference</code>	Critical values and p-values: "asymptotic" (standard normal, the default) or "t" (Student t with $G - 1$ degrees of freedom, a small-cluster reference convention in common use; it is not an exact finite-sample t law under arbitrary within-cluster dependence). The coefficient and standard error are identical under both; only the interval and p-value change.
<code>formula</code>	One of the two restricted formula layouts documented in cjive . Sections accept only additive lists of bare variable names, with exactly one outcome and one endogenous regressor; <code>0</code> or <code>-1</code> removes the intercept, while other arithmetic and subtraction are rejected.
<code>data</code>	A data frame in which to evaluate the formula.
<code>subset</code>	Optional expression selecting the rows to use, evaluated in <code>data</code> (formula methods only), as in lm .

`na.action` How to treat missing values (formula method only): `na.omit` (the default) drops every row with a missing value in any model component – outcome, endogenous regressor, instruments, controls, fixed effects, cluster or weights – in a single complete-cases filter; `na.fail` stops on any missing value; `na.pass` leaves them in, so the input validation reports them. Unlike the fitted-object interfaces, `iv_compare` returns a plain data frame and does not store or report the number of dropped rows. The default method (vector interface) does not handle missing values at all: it requires clean input and stops otherwise.

Details

The constructed instruments are: OLS, the residualised x itself; 2SLS, the full-sample fit $Z\hat{\pi}$; JIVE, the leave-one-out fit $(\hat{x} - hx)/(1 - h)$; CJIVE, the leave-cluster-out block fit. The CJIVE row equals `cjive(..., method = "dense")` on the same design. Weighting and residualisation precede validation: transformed values must be finite, y , x and each instrument must retain usable variation, and the residualised instrument matrix must have full numerical column rank. Absorbed or collinear instruments are not silently dropped; a numerically zero estimator denominator is rejected before division.

Because covariates (and the intercept) are partialled out by Frisch-Waugh-Lovell before the jackknife – the package-wide convention – the leverage h and the fit $\hat{x}$ in the "JIVE" row are computed on the *residualised* instruments. This is the improved JIVE (IJIVE) of Akerberg and Devereux (2009), not the original JIVE of Angrist, Imbens and Krueger (1999), which jackknifes the covariates alongside the instruments; residualising first removes the covariate-count bias that the original estimator carries, and the two coincide only when there are no covariates to partial out. The row keeps the historical label "JIVE" for backwards compatibility with clusterIV 0.1.0; FLM label the corresponding estimator IJIVE.

Value

A data frame with one row per estimator (in the order OLS, 2SLS, JIVE, CJIVE) and columns `estimator`, `coefficient`, `se`, `statistic`, `p.value`, `conf.low`, `conf.high`.

References

- Akerberg, D. A. and Devereux, P. J. (2009). Improved JIVE estimators for overidentified linear models with and without heteroskedasticity. *Review of Economics and Statistics*, 91(2), 351–362. The partial-out-then-jackknife construction of the "JIVE" row.
- Angrist, J. D., Imbens, G. W. and Krueger, A. B. (1999). Jackknife instrumental variables estimation. *Journal of Applied Econometrics*, 14(1), 57–67. The original JIVE that IJIVE improves upon.
- Frandsen, B., Leslie, E. and McIntyre, S. (2025). Cluster Jackknife Instrumental Variables Estimation. *Review of Economics and Statistics*.

See Also

`cjive` for the CJIVE fit with diagnostics; `iv_infer` for the recommended inference workflow. Two precomputed worked examples on real data: `vignette("queens-workflow", package = "clusterIV")` and `vignette("miami-bail", package = "clusterIV")`.

Examples

```
set.seed(2)
G <- 50; ng <- 5; n <- G * ng
cl <- rep(seq_len(G), each = ng)
z <- matrix(rnorm(n * 3), n, 3)
u <- rnorm(G)[cl]
x <- z %*% c(1, -1, 0.5) + u + rnorm(n)
y <- 2 * x + u + rnorm(n)
iv_compare(y, x, z, cluster = cl)
```

iv_infer

*Cluster jackknife IV estimation and weak-instrument-robust inference
in one call*

Description

Runs the recommended workflow of this package – the CJIVE point estimate ([cjive](#)), the CJAR confidence set ([cjar](#)), and the CJS test of a point null ([cjsscore](#)) – with the expensive preprocessing shared rather than repeated. The three help pages describe this workflow and previously required three calls; `iv_infer()` performs it with one data preparation, one Cholesky factorisation of the instrument Gram matrix, one whitened leave-cluster-out pass (fitted values and the `maxlev` leverage diagnostic from the same loop) and one set of per-cluster instrument sums shared by every selected test kernel. Every computational result agrees with the standalone calls on the same design.

Usage

```
iv_infer(y, ...)

## Default S3 method:
iv_infer(
  y,
  x,
  z,
  cluster,
  controls = NULL,
  fixed_effects = NULL,
  weights = NULL,
  intercept = TRUE,
  level = 0.95,
  beta0 = 0,
  calibration = c("chisq", "normal"),
  inference = c("asymptotic", "t"),
  tests = c("cjar", "cjsscore"),
  variance = c("plain", "crossfit"),
  ...
)
```

```

## S3 method for class 'formula'
iv_infer(
  formula,
  data,
  cluster,
  controls = NULL,
  fixed_effects = NULL,
  weights = NULL,
  subset,
  na.action = stats::na.omit,
  intercept = TRUE,
  level = 0.95,
  beta0 = 0,
  calibration = c("chisq", "normal"),
  inference = c("asymptotic", "t"),
  tests = c("cjar", "cjscore"),
  variance = c("plain", "crossfit"),
  ...
)

## S3 method for class 'iv_infer'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

```

Arguments

y	Outcome (numeric vector). Factor and character outcomes are rejected. The generic dispatches on this argument; a formula as the first argument selects the formula method (see formula).
...	Must be empty. Unknown, misspelled and unnamed arguments are errors.
x	Single endogenous regressor (numeric vector). Factor and character regressors are rejected; for the print methods, a fitted "cjive" object.
z	Instruments: a numeric vector/matrix, or a factor/character grouping vector for a judge design. A grouping instrument uses reference coding whenever an intercept or any fixed effect is partialled out. It uses one column for every level only when intercept = FALSE and no fixed effects are supplied.
cluster	Cluster identifiers (length n). In the formula methods all four forms work: a bare column name (cluster = cl), a one-sided formula (cluster = ~cl), a column name as a string (cluster = "cl"), or a full vector (cluster = df\$cl); a bare name is looked up in data first, then in the caller. A cluster formula must contain exactly one bare variable; multiway cluster formulas and arithmetic combinations of cluster codes are not supported.
controls	Optional covariates (FLM's X): a matrix or data frame, or a one-sided formula in the formula methods. May be rank deficient (fixed effects are allowed). An intercept is added unless intercept = FALSE.

fixed_effects	Optional high-dimensional fixed effects to absorb: a factor, or a list/data frame of factors (one per dimension). In the formula methods they can also enter through either formula layout or as a one-sided formula (<code>fixed_effects = ~ fe1 + fe2</code>). Absorbed by a matrix-free, rowsum-based joint projection before the dense partialling out; numerically identical (to tolerance) to entering the corresponding dummies via <code>controls</code> , but without forming any dummy matrix. Absorption spans the intercept within groups, and a result is returned only after normalized refinement and per-dimension weighted orthogonality checks pass.
weights	Optional numeric precision weights. Values must be finite and strictly positive; factor and character weights are rejected. In the formula methods the same four forms as <code>cluster</code> work, including a bare data-frame column name.
intercept	One non-missing logical value; partial out an intercept (default TRUE).
level	One finite numeric confidence level strictly between 0 and 1 for the CJIVE interval and CJAR/CJS confidence sets (default 0.95).
beta0	One finite numeric null value of the structural coefficient at which the CJAR and CJS statistics and p-values are reported (default 0).
calibration	CJAR critical-value calibration, "chisq" (the default) or "normal"; see cjar .
inference	CJIVE critical values and p-values, "asymptotic" (the default) or "t"; see cjive .
tests	Character vector selecting the test rows of the panel (the CJIVE estimate row is always present): any subset of "cjar" and "cjscore" (the cluster jackknife tests). When omitted, the recommended "cjar" and "cjscore" pair is selected. NULL and character(0) request a CJIVE-only panel. Values are uniquely resolved, deduplicated and stored in the canonical order shown above; unknown or ambiguous partial values are errors. Component fields always exist and are NULL when deselected.
variance	Variance estimator for selected CJAR/CJS components, "plain" (default) or "crossfit"; see cjar . The cross-fit calculation is shared when both components are selected and skipped when neither is selected.
formula	A model formula in either restricted layout documented in cjive : $y \sim x \mid z$ (optionally $y \sim x \mid z \mid fe$) or $y \sim exog \mid fe \mid endo \sim inst$. Formula sections are additive lists of bare variable names, with exactly one outcome and one endogenous regressor; arithmetic, transformed terms and subtraction are rejected except for the 0/-1 intercept markers.
data	A data frame in which to evaluate the formula.
subset	Optional expression selecting the rows to use, evaluated in <code>data</code> (formula methods only), as in lm .
na.action	How to treat missing values (formula methods only): na.omit (the default) drops every row with a missing value in any model component – outcome, endogenous regressor, instruments, controls, fixed effects, cluster or weights – in a single complete-cases filter, and records the count as <code>n_dropped</code> (reported by <code>print</code> when non-zero); <code>na.fail</code> stops on any missing value; <code>na.pass</code> leaves them in, so the input validation reports them. The default methods (vector interface) do not handle missing values at all: they require clean input and stop otherwise.

digits	Number of significant digits to print.
object	A fitted "iv_infer" object.

Details

Inputs follow the strict shared contract in [cjive](#). In particular, the weighted, residualised data must be finite; y , x and every instrument must retain usable variation after dense FWL or fixed-effect absorption; and the residualised instrument matrix must have full numerical column rank. Absorbed or collinear instrument columns are reported rather than silently dropped. The CJIVE component also stops before division when its IV denominator is numerically zero.

The shared work is preprocessing, not approximation: one data preparation, one instrument-Gram Cholesky factorisation, one whitened leave-cluster-out pass and one set of per-cluster instrument sums feed the CJIVE first stage, the CJAR quartic coefficients and the CJS coefficients; each selected test then performs its own coefficient and inversion work. Component results agree fieldwise with the standalone functions apart from `call`. The three advisories the selected jackknife tests issue (fewer than 20 clusters, a dominating cluster, within-cluster leverage near 1) fire once per `iv_infer()` call, not once per component; a CJIVE-only panel emits none of those test advisories.

Reporting conventions follow the component documentation: report the CJIVE coefficient for magnitude, the CJAR set for inference (it remains valid under weak and many instruments, unlike the CJIVE t-interval), and use the CJS p-value for the specific point null β_0 , where its complementary power properties are relevant.

Value

An object of class "iv_infer": a list with

`cjive`, `cjar`, `cjscore` objects of the three existing classes whose computational fields match the standalone calls `cjive()`, `cjar()` and `cjscore()` on the same design; only `call` records the enclosing panel invocation. Thus every existing method (`coef`, `confint`, `print`, `nobs`, ...) keeps working on the components. The CJIVE component always uses the dense path. `cjar` and `cjscore` are NULL when deselected via tests.

`n`, `G`, `k` observations, clusters, and number of instrument columns after partialling.

`maxlev` the shared within-cluster leverage diagnostic (see [cjar](#)).

`F_eff`, `K_eff`, `F_eff_crit`, `ng_max` the Montiel Olea-Pflueger effective first-stage F and its simplified-TSLS critical value (identical on every component), and the largest cluster size; see [cjar](#). If the estimated denominator is exactly zero with nonzero signal, `F_eff` = Inf and both `K_eff` and `F_eff_crit` are NA.

`fe_dims`, `fe_levels`, `k_controls`, `n_dropped` as in [cjive](#).

`tests` the resolved, canonical tests selector.

`term`, `beta0`, `level`, `calibration`, `inference`, `variance` the endogenous-regressor label and panel-wide reporting choices.

`call` the matched call.

Methods (by generic)

- `print(iv_infer)`: Print the panel: CJIVE point estimate and SE, the CJAR confidence set with its shape, the CJS p-value at β_0 , F_{CJ} against its critical value, and `maxlev`. Rows deselected via tests are omitted.
- `nobs(iv_infer)`: Number of observations used.

See Also

`cjive`, `cjar`, `cjscore` for the components and their full documentation (statistics, confidence set shapes, controls and fixed effects, caveats); `iv_compare` for estimator comparison. Two precomputed worked examples on real data: `vignette("queens-workflow", package = "clusterIV")` and `vignette("miami-bail", package = "clusterIV")`.

Examples

```
## Simulated judge design, as in ?cjar
set.seed(42)
G <- 30; ng <- 8; n <- G * ng
cl <- rep(seq_len(G), each = ng)      # cluster identifier
judge <- factor(rep(1:6, length.out = n)) # judge identity (grouping z)
u <- rnorm(G)[cl]                    # cluster-level shock
x <- 0.6 * as.numeric(judge) + u + rnorm(n)
y <- 0.5 * x + u + rnorm(n)

## One call: CJIVE estimate + CJAR set + CJS point-null test
fit <- iv_infer(y, x, judge, cluster = cl)
fit

## The components are ordinary fitted objects
coef(fit$cjive)
confint(fit$cjar)

## Formula interface, controls inside the formula
dat <- data.frame(y = y, x = x, judge = judge, cl = cl)
iv_infer(y ~ 1 | x ~ judge, data = dat, cluster = cl)
```

plot.cjar

Plot the p-value curve of a cluster jackknife test

Description

Draws the p-value $p(\beta)$ of the CJAR (or CJS) test as a function of the hypothesised coefficient, with the reported confidence set shaded, a horizontal line at $\alpha = 1 - \text{level}$, and β_0 marked. The curve is evaluated from the polynomial coefficients stored on the object, so no refitting occurs; the shaded regions are taken from `x$conf_set` – never re-derived from the plotted grid – so the picture cannot silently disagree with the reported set. An unbounded set stops looking like a failure here: it is a curve that never dips below the threshold in the tails (Dufour 1997).

Usage

```
## S3 method for class 'cjar'
plot(
  x,
  from = NULL,
  to = NULL,
  n = 512L,
  main = "CJAR p-value curve",
  xlab = expression(beta),
  ylab = "p-value",
  ylim = c(0, 1),
  col = "black",
  lty = 1L,
  lwd = 1.5,
  shade.col = "grey88",
  shade.border = NA,
  alpha.col = "black",
  alpha.lty = 3L,
  ...
)

## S3 method for class 'cjscore'
plot(
  x,
  from = NULL,
  to = NULL,
  n = 512L,
  main = "CJS p-value curve",
  xlab = expression(beta),
  ylab = "p-value",
  ylim = c(0, 1),
  col = "black",
  lty = 1L,
  lwd = 1.5,
  shade.col = "grey88",
  shade.border = NA,
  alpha.col = "black",
  alpha.lty = 3L,
  ...
)

## S3 method for class 'iv_infer'
plot(
  x,
  from = NULL,
  to = NULL,
  n = 512L,
  main = NULL,
```

```

xlab = expression(beta),
ylab = "p-value",
ylim = c(-0.1, 1),
col = c("black", "steelblue4"),
lty = c(1L, 2L),
lwd = 1.5,
shade.col = "grey88",
shade.border = NA,
alpha.col = "black",
alpha.lty = 3L,
...
)

```

Arguments

<code>x</code>	A fitted "cjar", "cjscore" or "iv_infer" object.
<code>from, to</code>	When supplied, finite numeric scalar plot limits satisfying <code>from < to</code> . Each omitted limit is taken from a range determined by the stored coefficients: the finite endpoints of the confidence set are padded by half their width when at least two distinct ones exist. Otherwise the range is centred on the CJAR numerator-quadratic vertex or the CJS score root, with four times the coefficient-based scale on each side $((w_0/w_4)^{1/4}$ or $\sqrt{v_0/v_2}$ when the required coefficients are positive, and 1 otherwise), then widened to include any single finite endpoint.
<code>n</code>	Finite integer-like number of grid points, at least 2 (default 512).
<code>main</code>	Plot title. For an <code>iv_infer</code> object, NULL chooses a title naming the test components actually present.
<code>xlab, ylab</code>	Axis labels.
<code>ylim</code>	Two finite increasing y-axis limits.
<code>col, lty, lwd</code>	Curve colour, line type, and line width; vectors are recycled when the panel contains two curves.
<code>shade.col, shade.border</code>	Fill and border for accepted regions.
<code>alpha.col, alpha.lty</code>	Colour and line type of the test-size line.
<code>...</code>	Additional named arguments passed to <code>plot</code> . Every argument must be named. Use <code>from/to</code> , not <code>xlim</code> ; <code>x</code> , <code>y</code> , <code>xlim</code> , and <code>type</code> conflict with the p-value-curve contract and are rejected.

Details

`plot.iv_infer` plots every available jackknife test curve. With CJAR present its confidence set supplies the shading; with CJS alone, the CJS set supplies it. The title and legend identify the plotted tests. Positive-width accepted intervals are shaded and isolated accepted points (degenerate `[b, b]` rows) are drawn as vertical marks. The CJIVE point estimate and the visible part of its Wald interval are marked beneath the curves when they overlap the requested horizontal range.

Value

Invisibly, x.

Examples

```
set.seed(42)
G <- 30; ng <- 8; n <- G * ng
cl <- rep(seq_len(G), each = ng)
judge <- factor(rep(1:6, length.out = n))
u <- rnorm(G)[cl]
x <- 0.6 * as.numeric(judge) + u + rnorm(n)
y <- 0.5 * x + u + rnorm(n)
plot(cjar(y, x, judge, cluster = cl))
plot(iv_infer(y, x, judge, cluster = cl))
```

Index

cjar, [2](#), [16–24](#), [27](#), [30](#), [32–34](#)
cjive, [2–5](#), [8](#), [9](#), [11](#), [12](#), [19](#), [20](#), [22–24](#), [27–30](#),
[32–34](#)
cjscore, [6](#), [11](#), [17](#), [18](#), [27](#), [30](#), [34](#)
clusterIV-tidiers, [25](#)
coef.cjive (cjive), [12](#)
confint.cjar (cjar), [2](#)
confint.cjive (cjive), [12](#)
confint.cjscore (cjscore), [18](#)

glance.cjar (clusterIV-tidiers), [25](#)
glance.cjive (clusterIV-tidiers), [25](#)
glance.cjscore (clusterIV-tidiers), [25](#)
glance.iv_infer (clusterIV-tidiers), [25](#)

iv_compare, [11](#), [15](#), [17](#), [24](#), [27](#), [34](#)
iv_infer, [11](#), [16](#), [17](#), [24](#), [27](#), [29](#), [30](#)

lm, [5](#), [14](#), [20](#), [28](#), [32](#)

na.omit, [14](#), [29](#), [32](#)
nobs.cjar (cjar), [2](#)
nobs.cjive (cjive), [12](#)
nobs.cjscore (cjscore), [18](#)
nobs.iv_infer (iv_infer), [30](#)

plot, [36](#)
plot.cjar, [34](#)
plot.cjscore (plot.cjar), [34](#)
plot.iv_infer (plot.cjar), [34](#)
print.cjar (cjar), [2](#)
print.cjive (cjive), [12](#)
print.cjscore (cjscore), [18](#)
print.iv_infer (iv_infer), [30](#)
print.summary.cjar (cjar), [2](#)
print.summary.cjive (cjive), [12](#)
print.summary.cjscore (cjscore), [18](#)

summary.cjar, [23](#)
summary.cjar (cjar), [2](#)
summary.cjive (cjive), [12](#)

summary.cjscore (cjscore), [18](#)

tidy.cjar (clusterIV-tidiers), [25](#)
tidy.cjive (clusterIV-tidiers), [25](#)
tidy.cjscore (clusterIV-tidiers), [25](#)
tidy.iv_infer (clusterIV-tidiers), [25](#)

vcov.cjive (cjive), [12](#)