# Package 'RPIV'

March 24, 2026

**Title** Residual Prediction Tests for Well-Specification of Instrumental Variable Models

**Version** 1.1.0

**Description** Two tests for the well-specification of the linear instrumental variable model. The first test is based on trying to predict the residuals of a two-stage least-squares regression using a random forest. The second test is robust to weak-identification and based on trying to predict the residuals for a particular candidate parameter and can also be used to construct confidence sets with an Anderson-Rubin-type inversion. Details can be found in Scheidegger, Londschien and Bühlmann (2025) ``Machine-learning-powered specification testing in linear instrumental variable models'' <doi:10.48550/arXiv.2506.12771>.

**URL** https://github.com/cyrillsch/RPIV

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Imports** ranger, stats

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Cyrill Scheidegger [aut, cre, cph] (ORCID: <https://orcid.org/0009-0005-2851-1384>)

**Maintainer** Cyrill Scheidegger <cyrill.scheidegger@stat.math.ethz.ch>

**Repository** CRAN

**Date/Publication** 2026-03-24 10:20:09 UTC

## Contents

RPIV_test *Residual Prediction Test for Linear Instrumental Variable Models*

## Description

Performs a hypothesis test for the well-specification of linear instrumental variable (IV) model. More specifically, it tests the null-hypothesis $H_0 : \exists \beta \in \mathbb{R}^p$ s.t. $\mathbb{E}[Y - X^T\beta|Z] = 0$. It uses sample splitting and a random forest to try to predict the two-stage least-squares residuals from the instruments $Z$.

## Usage

```
RPIV_test(
  Y,
  X,
  C = NULL,
  Z,
  frac_A = NULL,
  gamma = 0.05,
  variance_estimator = "heteroskedastic",
  clustering = NULL,
  upper_clip_quantile = 0.8,
  regr_par = list(),
  fit_intercept = TRUE
)
```

## Arguments

| | |
|---|---|
| Y | A numeric vector. The outcome variable. |
| X | A numeric matrix or vector. The endogenous explanatory variables. |
| C | A numeric matrix, vector or NULL. The additional exogenous explanatory variables (optional). |
| Z | A numeric matrix or vector. The instruments. |
| frac_A | A numeric scalar between 0 and 1 or NULL. The fraction of the sample used for training (sample splitting). Default is min(0.5, exp(1)/log(n)), where n is the sample size. |
| gamma | A non-negative scalar. If the variance estimator is less than gamma times the noise level (as estimated as by the mean of the squared residuals), gamma times the noise level is used as variance estimator. |
| variance_estimator | Character string or vector. One or more of "homoskedastic", "heteroskedastic", "cluster". Specifies the types of variance estimation used. |
| clustering | A vector of cluster identifiers or NULL. Observations with the same value of clustering belong to the same cluster. Required if variance_estimator includes "cluster". |

`upper_clip_quantile`
: A scalar between 0 and 1. The estimated weight-function will be clipped at the corresponding quantile of the random forest predictions on the auxiliary sample. Use 0 to use the sign of the predictions. Default is 0.8.

`regr_par`
: A list of parameters passed to the random forest regression model. Supports `num.trees`, `num_mtry` (number of different mtry values to try out) or a vector `mtry`, a vector `max.depth`, `num_min.node.size` (number of different min.node.size values to try out) or a vector `min.node.size`.

`fit_intercept`
: Logical. Should an intercept be included in the model? Default is `TRUE`.

## Details

The RPIV test splits the sample into an auxiliary and a main sample. On the auxiliary sample, a random forest is used to predict the two-stage least squares residuals from the instruments. The test statistic is the scalar product of the two-stage least-squares residuals with a clipped and rescaled version of the learned function evaluated on the main sample divided by an estimator of its standard deviation.

If `clustering` is supplied, sample splitting is done at cluster level (also for `variance_estimator` `"homoskedastic"` or `"heteroskedastic"`).

## Value

If a single variance estimator is used, returns a list with:

**p_value** p-value of the residual prediction test.

**test_statistic** The value of the test statistic.

**var_fraction** The estimated variance fraction, i.e., variance estimator divided by noise level estimate.

**T_null** The value of the initial test statistic. If var_fraction >= gamma, it is equal to test_statistic, otherwise, it has larger absolute value.

**variance_estimator** The variance estimator used.

If multiple estimators are supplied, returns a named list of such results for each estimator.

## References

Cyrill Scheidegger, Malte Londschien and Peter Bühlmann. Machine-learning-powered specification testing in linear instrumental variable models. Preprint, doi:10.48550/arXiv.2506.12771, 2025.

## Examples

```
set.seed(1)
n <- 100
Z <- rnorm(n)
H <- rnorm(n)
C <- rnorm(n)
X <- Z + rnorm(n) + H
Y1 <- X - C - H + rnorm(n)
```

```
Y2 <- X - C - H + rnorm(n) + Z^2
RPIV_test(Y1, X, C, Z)
RPIV_test(Y2, X, C, Z)
```

---

`weak_RPIV_test` *Weak-IV-Robust Residual Prediction Test for Linear Instrumental Variable Models*

---

## Description

Constructs a hypothesis test for the well-specification of linear instrumental variable (IV) model at a particular value of the parameter. More specifically, it tests the null-hypothesis $H_0(\beta_0) : \mathbb{E}[Y - X^T\beta_0|Z] = 0$. It uses sample splitting and a random forest to try to predict the residuals from the instruments $Z$. The testing procedure remains valid under weak identification. The function returns a closure that must be evaluated at a candidate parameter value.

## Usage

```
weak_RPIV_test(
  Y,
  X,
  C = NULL,
  Z,
  frac_A = NULL,
  gamma = 0.05,
  variance_estimator = "heteroskedastic",
  clustering = NULL,
  upper_clip_quantile = 0.8,
  regr_par = list(),
  fit_intercept = TRUE,
  fit_at_tsls = TRUE,
  use_C_for_prediction = TRUE
)
```

## Arguments

| | |
|---|---|
| `Y` | A numeric vector. The outcome variable. |
| `X` | A numeric matrix or vector. The endogenous explanatory variables. |
| `C` | A numeric matrix, vector or `NULL`. The additional exogenous explanatory variables (optional). |
| `Z` | A numeric matrix or vector. The instruments. |
| `frac_A` | A numeric scalar between 0 and 1 or `NULL`. The fraction of the sample used for training (sample splitting). Default is `min(0.5, exp(1)/log(n))`, where `n` is the sample size. |

`gamma` A non-negative scalar. If the variance estimator is less than gamma times the noise level (as estimated as by the mean of the squared residuals), gamma times the noise level is used as variance estimator.

`variance_estimator` Character string or vector. One or more of `"homoskedastic"`, `"heteroskedastic"`, `"cluster"`. Specifies the types of variance estimation used.

`clustering` A vector of cluster identifiers or `NULL`. Observations with the same value of `clustering` belong to the same cluster. Required if `variance_estimator` includes `"cluster"`.

`upper_clip_quantile` A scalar between 0 and 1. The estimated weight-function will be clipped at the corresponding quantile of the random forest predictions on the auxiliary sample. Use 0 to use the sign of the predictions. Default is 0.8.

`regr_par` A list of parameters passed to the random forest regression model. Supports `num.trees`, `num_mtry` (number of different mtry values to try out) or a vector `mtry`, a vector `max.depth`, `num_min.node.size` (number of different min.node.size values to try out) or a vector `min.node.size`.

`fit_intercept` Logical. Should an intercept be included in the model (added to C)? Default is `TRUE`.

`fit_at_tsls` Logical. If `TRUE`, a random forest is initially tuned and fitted at the TSLS estimator. Subsequent evaluations may reuse this tuning or the learned partition structure. Default is `TRUE`.

`use_C_for_prediction` Logical. If `TRUE`, the random forest uses both `Z` and `C` as predictors. If `FALSE`, only Z is used. Default is `TRUE`.

## Details

The procedure is based on orthogonalized residuals and sample splitting. A random forest is trained on an auxiliary sample to predict structural residuals from the instruments. The resulting weight function is then partialled out with respect to exogenous covariates and evaluated on the main sample.

Sample splitting is performed either observation-wise or at the cluster level. Residuals and weight functions are orthogonalized with respect to exogenous covariates to ensure validity under weak identification.

## Value

A function `f(beta, type)` that computes the weak RPIV test statistic.

- `beta` is a numeric vector of coefficients.
- `type` determines how the weight function is constructed:

  `"tune_and_fit"` Re-tunes and refits the random forest.

  `"fit"` Uses tuning parameters obtained at TSLS to fit a new random forest.

  `"recalculate"` Reuses the partition obtained from the random forest with at the TSLS-residuals and recomputes leaf means.

The returned value is a named numeric vector with one entry per variance estimator. The resulting test statistic asymptotically follows a standard Gaussian distribution. P-values can be obtained from the test statistic `T_stat` as `1 - pnorm(T_stat)`.

## References

Cyrill Scheidegger, Malte Londschien and Peter Bühlmann. Machine-learning-powered specification testing in linear instrumental variable models. Preprint, doi:10.48550/arXiv.2506.12771, 2025.

## Examples

```
set.seed(1)
n <- 100
Z <- rnorm(n)
H <- rnorm(n)
C <- rnorm(n)
X <- Z + rnorm(n) + H
Y1 <- X - C - H + rnorm(n)
Y2 <- X - C - H + rnorm(n) + Z^2
test_statistic1 <- weak_RPIV_test(Y1, X, C, Z)
test_statistic1(1, "tune_and_fit")
test_statistic1(1, "fit")
test_statistic1(1, "recalculate")
test_statistic1(0, "tune_and_fit")
test_statistic1(0, "fit")
test_statistic1(0, "recalculate")
test_statistic2 <- weak_RPIV_test(Y2, X, C, Z)
test_statistic2(1, "tune_and_fit")
test_statistic2(1, "fit")
test_statistic2(1, "recalculate")
```

# Index