# Package 'ivreg'

February 9, 2026

**Title** Instrumental-Variables Regression by '2SLS', '2SM', or '2SMM', with Diagnostics

**Version** 0.6-6

**Date** 2026-02-09

**Description** Instrumental variable estimation for linear models by two-stage least-squares (2SLS) regression or by robust-regression via M-estimation (2SM) or MM-estimation (2SMM). The main ivreg() model-fitting function is designed to provide a workflow as similar as possible to standard lm() regression. A wide range of methods is provided for fitted ivreg model objects, including extensive functionality for computing and graphing regression diagnostics in addition to other standard model tools.

**License** GPL (>= 2)

**Depends** R (>= 3.6.0)

**Imports** car (>= 3.0-9), Formula, lmtest, MASS, stats

**Suggests** AER, effects (>= 4.2.0), knitr, insight, parallel, rmarkdown, sandwich, testthat, modelsummary, gt, ggplot2

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**BugReports** https://github.com/zeileis/ivreg/issues/

**URL** https://zeileis.github.io/ivreg/

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** John Fox [aut] (ORCID: <https://orcid.org/0000-0002-1196-8012>), Christian Kleiber [aut] (ORCID: <https://orcid.org/0000-0002-6781-4733>), Achim Zeileis [aut, cre] (ORCID: <https://orcid.org/0000-0003-0918-3766>), Nikolas Kuschnig [ctb] (ORCID: <https://orcid.org/0000-0002-6642-2543>), R Core Team [ctb]

**Maintainer** Achim Zeileis <Achim.Zeileis@R-project.org>

**Repository** CRAN

**Date/Publication** 2026-02-09 06:10:28 UTC

# Contents



---

`CigaretteDemand` *U.S. Cigarette Demand Data*

---

## Description

Determinants of cigarette demand for the 48 continental US States in 1995 and compared between 1995 and 1985.

## Usage

```
data("CigaretteDemand", package = "ivreg")
```

## Format

A data frame with 48 rows and 10 columns.

**packs** Number of cigarette packs per capita sold in 1995.

**rprice** Real price in 1995 (including sales tax).

**rincome** Real per capita income in 1995.

**salestax** Sales tax in 1995.

**cigtax** Cigarette-specific taxes (federal and average local excise taxes) in 1995.

**packsdiff** Difference in `log(packs)` (between 1995 and 1985).

**pricediff** Difference in `log(rprice)` (between 1995 and 1985).

**incomediff** Difference in `log(rincome)` (between 1995 and 1985).

**salestaxdiff** Difference in `salestax` (between 1995 and 1985).

**cigtaxdiff** Difference in `cigtax` (between 1995 and 1985).

## Details

The data are taken from the online complements to Stock and Watson (2007) and had been prepared as panel data (in long form) in `CigarettesSW` from the AER package (Kleiber and Zeileis 2008). Here, the data are provided by state (in wide form), readily preprocessed to contain all variables needed for illustrations of OLS and IV regressions. More related examples from Stock and Watson (2007) are provided in the AER package in `StockWatson2007`. A detailed discussion of the various cigarette demand examples with R code is provided by Hanck et al. (2020, Chapter 12).

## Source

Online complements to Stock and Watson (2007).

## References

Hanck, C., Arnold, M., Gerber, A., and Schmelzer, M. (2020). *Introduction to Econometrics with R*. https://www.econometrics-with-r.org/

Kleiber, C. and Zeileis, A. (2008). *Applied Econometrics with R*. Springer-Verlag

Stock, J.H. and Watson, M.W. (2007). *Introduction to Econometrics*, 2nd ed., Addison Wesley.

## See Also

CigarettesSW.

## Examples

```
## load data
data("CigaretteDemand", package = "ivreg")

## basic price elasticity: OLS vs. IV
cig_ols <- lm(log(packs) ~ log(rprice), data = CigaretteDemand)
cig_iv <- ivreg(log(packs) ~ log(rprice) | salestax, data = CigaretteDemand)
cbind(OLS = coef(cig_ols), IV = coef(cig_iv))

## adjusting for income differences (exogenous)
cig_iv2 <- ivreg(log(packs) ~ log(rprice) + log(rincome) | salestax + log(rincome),
  data = CigaretteDemand)
## adding a second instrument for log(rprice)
cig_iv3 <- update(cig_iv2, . ~ . | . + cigtax)

## comparison using heteroscedasticity-consistent standard errors
library("lmtest")
library("sandwich")
coeftest(cig_iv2, vcov = vcovHC, type = "HC1")
coeftest(cig_iv3, vcov = vcovHC, type = "HC1")

## long-run price elasticity using differences between 1995 and 1985
cig_ivdiff1 <- ivreg(packsdiff ~ pricediff + incomediff | incomediff + salestaxdiff,
  data = CigaretteDemand)
cig_ivdiff2 <- update(cig_ivdiff1, . ~ . | . - salestaxdiff + cigtaxdiff)
cig_ivdiff3 <- update(cig_ivdiff1, . ~ . | . + cigtaxdiff)
coeftest(cig_ivdiff1, vcov = vcovHC, type = "HC1")
coeftest(cig_ivdiff2, vcov = vcovHC, type = "HC1")
coeftest(cig_ivdiff3, vcov = vcovHC, type = "HC1")
```

coef.ivreg *Methods for* "ivreg" *Objects*

## Description

Various methods for processing "ivreg" objects; for diagnostic methods, see ivregDiagnostics.

## Usage

```
## S3 method for class 'ivreg'
coef(object, component = c("stage2", "stage1"), complete = TRUE, ...)

## S3 method for class 'ivreg'
vcov(object, component = c("stage2", "stage1"), complete = TRUE, ...)

## S3 method for class 'ivreg'
bread(x, ...)

## S3 method for class 'ivreg'
estfun(x, ...)

## S3 method for class 'ivreg'
vcovHC(x, component = "stage2", ...)

## S3 method for class 'ivreg'
terms(x, component = c("regressors", "instruments", "full"), ...)

## S3 method for class 'ivreg'
model.matrix(
  object,
  component = c("regressors", "projected", "instruments"),
  ...
)

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

## S3 method for class 'ivreg'
predict(
  object,
  newdata,
  type = c("response", "terms"),
  na.action = na.pass,
  se.fit = FALSE,
  interval = c("none", "confidence", "prediction"),
  df = Inf,
  level = 0.95,
```

```
  weights,
  ...
)

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

## S3 method for class 'ivreg'
update(object, formula., ..., evaluate = TRUE)

## S3 method for class 'ivreg'
residuals(
  object,
 type = c("response", "projected", "regressors", "working", "deviance", "pearson",
    "partial", "stage1"),
  ...
)

## S3 method for class 'ivreg'
Effect(focal.predictors, mod, ...)

## S3 method for class 'ivreg'
formula(x, component = c("complete", "regressors", "instruments"), ...)

## S3 method for class 'ivreg'
find_formula(x, ...)

## S3 method for class 'ivreg'
alias(object, ...)

## S3 method for class 'ivreg'
qr(x, ...)

## S3 method for class 'ivreg'
weights(object, type = c("working", "variance", "robustness"), ...)
```

## Arguments

| | |
|---|---|
| object, model, mod | An object of class "ivreg". |
| component | For terms, "regressors", "instruments", or "full"; for model.matrix, "projected", "regressors", or "instruments"; for formula, "regressors", "instruments", or "complete"; for coef and vcov, "stage2" or "stage1". |
| complete | If TRUE, the default, the returned coefficient vector (for coef) or coefficient-covariance matrix (for vcov) includes elements for aliased regressors. |
| ... | arguments to pass down. |
| x | An object of class "ivreg". |

| | |
|---|---|
| newdata | Values of predictors for which to obtain predicted values; if missing predicted (i.e., fitted) values are computed for the data to which the model was fit. |
| type | For predict, one of "response" (the default) or "terms"; for residuals, one of "response" (the default), "projected", "regressors", "working", "deviance", "pearson", or "partial"; type = "working" and "response" are equivalent, as are type = "deviance" and "pearson"; for weights, "working" (or equivalently "variance", the default) for invariance-variance weights (which is NULL for an unweighted fit) or "robustness" for robustness weights (available for M or MM estimation). |
| na.action | na method to apply to predictor values for predictions; default is na.pass. |
| se.fit | Compute standard errors of predicted values (default FALSE). |
| interval | Type of interval to compute for predicted values: "none" (the default), "confidence" for confidence intervals for the expected response, or "prediction" for prediction intervals for future observations. |
| df | For predict, degrees of freedom for computing t-distribution confidence- or prediction-interval limits; the default, Inf, is equivalent to using the normal distribution; if NULL, df is taken from the residual degrees of freedom for the model. These tests are not to be confused with the *regression diagnostics* provided elsewhere in the **ivreg** package: see ivregDiagnostics. |
| level | for confidence or prediction intervals, default 0.95. |
| weights | Either a numeric vector or a one-sided formula to provide weights for prediction intervals when the fit is weighted. If weights and newdata are missing, the weights are those used for fitting the model. |
| digits | For printing. |
| formula. | To update model. |
| evaluate | If TRUE, the default, the updated model is evaluated; if FALSE the updated call is returned. |
| focal.predictors | Focal predictors for effect plot, see Effect. |

## See Also

ivreg, ivreg.fit, ivregDiagnostics

---

confint.ivreg *Summary and Inference Methods for "ivreg" Objects*

---

## Description

Summary method, including Wald tests and (by default) certain diagnostic tests, for "ivreg" model objects, as well as other related inference functions.

## Usage

```
## S3 method for class 'ivreg'
confint(
  object,
  parm,
  level = 0.95,
  component = c("stage2", "stage1"),
  complete = TRUE,
  vcov. = NULL,
  df = NULL,
  ...
)

## S3 method for class 'ivreg'
summary(object, vcov. = NULL, df = NULL, diagnostics = NULL, ...)

## S3 method for class 'summary.ivreg'
print(
  x,
  digits = max(3, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)

## S3 method for class 'ivreg'
anova(object, object2, test = "F", vcov. = NULL, ...)

## S3 method for class 'ivreg'
Anova(mod, test.statistic = c("F", "Chisq"), vcov. = NULL, ...)

## S3 method for class 'ivreg'
linearHypothesis(
  model,
  hypothesis.matrix,
  rhs = NULL,
  test = c("F", "Chisq"),
  vcov. = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| object, object2, model, mod | An object of class "ivreg". |
| parm | parameters for which confidence intervals are to be computed; a vector or numbers or names; the default is all parameters. |
| level | confidence level; the default is 0.95. |
| component | Character indicating "stage2" or "stage1". |

| | |
|---|---|
| complete | If TRUE, the default, the returned coefficient vector (for coef) or coefficient-covariance matrix (for vcov) includes elements for aliased regressors. |
| vcov. | Optionally either a coefficient covariance matrix or a function to compute such a covariance matrix from fitted ivreg model objects. If NULL (the default) the standard covariance matrix (based on the information matrix) is used. Alternatively, covariance matrices (e.g., clustered and/or heteroscedasticity-consistent) can be plugged in to adjust Wald tests or confidence intervals etc. In summary, if diagnostics = TRUE, vcov. must be a function (not a matrix) because the alternative covariances are also needed for certain auxiliary models in the diagnostic tests. If vcov. is a function, the ... argument can be used to pass on further arguments to this function. |
| df | For summary, optional residual degrees of freedom to use in computing model summary. |
| ... | arguments to pass down. |
| diagnostics | Report 2SLS "diagnostic" tests in model summary (default is TRUE). These tests are not to be confused with the *regression diagnostics* provided elsewhere in the **ivreg** package: see ivregDiagnostics. |
| x | An object of class "summary.ivreg". |
| digits | Minimal number of significant digits for printing. |
| signif.stars | Show "significance stars" in summary output? |
| test, test.statistic | Test statistics for ANOVA table computed by anova, Anova, or linearHypothesis. Only test = "F" is supported by anova; this is also the default for Anova and linearHypothesis, which also allow test = "Chisq" for asymptotic tests. |
| hypothesis.matrix, rhs | For formulating a linear hypothesis; see the documentation for linearHypothesis for details. |

## See Also

ivreg, ivreg.fit, ivregDiagnostics

## Examples

```
## data and model
data("CigaretteDemand", package = "ivreg")
m <- ivreg(log(packs) ~ log(rincome) | log(rprice) | salestax, data = CigaretteDemand)

## summary including diagnostics
summary(m)

## replicate global F test from summary (against null model) "by hand"
m0 <- ivreg(log(packs) ~ 1, data = CigaretteDemand)
anova(m0, m)

## or via linear hypothesis test
car::linearHypothesis(m, c("log(rincome)", "log(rprice)"))
```

```
## confidence intervals
confint(m)

## just the Wald tests for the coefficients
library("lmtest")
coeftest(m)

## plug in a heteroscedasticity-consistent HC1 covariance matrix (from sandwich)
library("sandwich")
## - as a function passing additional type argument through ...
coeftest(m, vcov = vcovHC, type = "HC1")
## - as a function without additional arguments
hc1 <- function(object, ...) vcovHC(object, type = "HC1", ...)
coeftest(m, vcov = hc1)
## - as a matrix
vc1 <- vcovHC(m, type = "HC1")
coeftest(m, vcov = vc1)

## in summary() with diagnostics = TRUE use one of the function specifications,
## the matrix is only possible when diagnostics = FALSE
summary(m, vcov = vcovHC, type = "HC1")     ## function + ...
summary(m, vcov = hc1)                      ## function
summary(m, vcov = vc1, diagnostics = FALSE) ## matrix

## in confint() and anova() any of the three specifications can be used
anova(m0, m, vcov = vcovHC, type = "HC1")   ## function + ...
anova(m0, m, vcov = hc1)                    ## function
anova(m0, m, vcov = vc1)                    ## matrix
```

---

`influence.ivreg` *Deletion and Other Diagnostic Methods for* `"ivreg"` *Objects*

---

## Description

Methods for computing deletion and other regression diagnostics for 2SLS regression. It's generally more efficient to compute the deletion diagnostics via the `influence` method and then to extract the various specific diagnostics with the methods for `"influence.ivreg"` objects. Other diagnostics for linear models, such as added-variable plots (`avPlots`) and component-plus-residual plots (`crPlots`), also work, as do effect plots (e.g., `predictorEffects`) with residuals (see the examples below). The pointwise confidence envelope for the `qqPlot` method assumes an independent random sample from the t distribution with degrees of freedom equal to the residual degrees of freedom for the model and so are approximate, because the studentized residuals aren't independent.

For additional information, see the vignette Diagnostics for 2SLS Regression.

## Usage

```
## S3 method for class 'ivreg'
influence(
```

```
  model,
  sigma. = n <= 1000,
  type = c("stage2", "both", "maximum"),
  applyfun = NULL,
  ncores = NULL,
  ...
)

## S3 method for class 'ivreg'
rstudent(model, ...)

## S3 method for class 'ivreg'
cooks.distance(model, ...)

## S3 method for class 'influence.ivreg'
dfbeta(model, ...)

## S3 method for class 'ivreg'
dfbeta(model, ...)

## S3 method for class 'ivreg'
hatvalues(model, type = c("stage2", "both", "maximum", "stage1"), ...)

## S3 method for class 'influence.ivreg'
rstudent(model, ...)

## S3 method for class 'influence.ivreg'
hatvalues(model, ...)

## S3 method for class 'influence.ivreg'
cooks.distance(model, ...)

## S3 method for class 'influence.ivreg'
qqPlot(
  x,
  ylab = paste("Studentized Residuals(", deparse(substitute(x)), ")", sep = ""),
  distribution = c("t", "norm"),
  ...
)

## S3 method for class 'ivreg'
influencePlot(model, ...)

## S3 method for class 'influence.ivreg'
influencePlot(model, ...)

## S3 method for class 'ivreg'
infIndexPlot(model, ...)
```

```
## S3 method for class 'influence.ivreg'
infIndexPlot(model, ...)

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

## S3 method for class 'ivreg'
avPlots(model, terms, ...)

## S3 method for class 'ivreg'
avPlot(model, ...)

## S3 method for class 'ivreg'
mcPlots(model, terms, ...)

## S3 method for class 'ivreg'
mcPlot(model, ...)

## S3 method for class 'ivreg'
Boot(
  object,
  f = coef,
  labels = names(f(object)),
  R = 999,
  method = "case",
  ncores = 1,
  ...
)

## S3 method for class 'ivreg'
crPlots(model, terms, ...)

## S3 method for class 'ivreg'
crPlot(model, ...)

## S3 method for class 'ivreg'
ceresPlots(model, terms, ...)

## S3 method for class 'ivreg'
ceresPlot(model, ...)

## S3 method for class 'ivreg'
plot(x, ...)

## S3 method for class 'ivreg'
qqPlot(x, distribution = c("t", "norm"), ...)
```

```
## S3 method for class 'ivreg'
outlierTest(model, ...)

## S3 method for class 'ivreg'
spreadLevelPlot(x, main = "Spread-Level Plot", ...)

## S3 method for class 'ivreg'
ncvTest(model, ...)

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

## S3 method for class 'rivreg'
influence(model, ...)
```

## Arguments

| | |
|---|---|
| model, x, object | A "ivreg" or "influence.ivreg" object. |
| sigma. | If TRUE (the default for 1000 or fewer cases), the deleted value of the residual standard deviation is computed for each case; if FALSE, the overall residual standard deviation is used to compute other deletion diagnostics. |
| type | If "stage2" (the default), hatvalues are for the second stage regression; if "both", the hatvalues are the geometric mean of the casewise hatvalues for the two stages; if "maximum", the hatvalues are the larger of the casewise hatvalues for the two stages. In computing the geometric mean or casewise maximum hatvalues, the hatvalues for each stage are first divided by their average (number of coefficients in stage regression/number of cases); the geometric mean or casewise maximum values are then multiplied by the average hatvalue from the second stage. |
| applyfun | Optional loop replacement function that should work like lapply with arguments function(X, FUN, ...). The default is to use a loop unless the ncores argument is specified (see below). |
| ncores | Numeric, number of cores to be used in parallel computations. If set to an integer the applyfun is set to use either parLapply (on Windows) or mclapply (otherwise) with the desired number of cores. |
| ... | arguments to be passed down. |
| ylab | The vertical axis label. |
| distribution | "t" (the default) or "norm". |
| terms | Terms for which added-variable plots are to be constructed; the default, if the argument isn't specified, is the "regressors" component of the model formula. |
| f, labels, R | see Boot. |
| method | only "case" (case resampling) is supported: see Boot. |
| main | Main title for the graph. |

## Value

In the case of `influence.ivreg`, an object of class `"influence.ivreg"` with the following components:

`coefficients` the estimated regression coefficients

`model` the model matrix

`dfbeta` influence on coefficients

`sigma` deleted values of the residual standard deviation

`dffits` overall influence on the regression coefficients

`cookd` Cook's distances

`hatvalues` hatvalues

`rstudent` Studentized residuals

`df.residual` residual degrees of freedom

In the case of other methods, such as `rstudent.ivreg` or `rstudent.influence.ivreg`, the corresponding diagnostic statistics. Many other methods (e.g., `crPlot.ivreg`, `avPlot.ivreg`, `Effect.ivreg`) draw graphs.

## See Also

`ivreg`, `avPlots`, `crPlots`, `predictorEffects`, `qqPlot`, `influencePlot`, `infIndexPlot`, `Boot`, `outlierTest`, `spreadLevelPlot`, `ncvTest`.

## Examples

```
kmenta.eq1 <- ivreg(Q ~ P + D | D + F + A, data = Kmenta)
summary(kmenta.eq1)
car::avPlots(kmenta.eq1)
car::mcPlots(kmenta.eq1)
car::crPlots(kmenta.eq1)
car::ceresPlots(kmenta.eq1)
car::influencePlot(kmenta.eq1)
car::influenceIndexPlot(kmenta.eq1)
car::qqPlot(kmenta.eq1)
car::spreadLevelPlot(kmenta.eq1)
plot(effects::predictorEffects(kmenta.eq1, residuals = TRUE))
set.seed <- 12321 # for reproducibility
confint(car::Boot(kmenta.eq1, R = 250)) # 250 reps for brevity
car::outlierTest(kmenta.eq1)
car::ncvTest(kmenta.eq1)
```

ivreg *Instrumental-Variable Regression by 2SLS, 2SM, or 2SMM Estimation*

## Description

Fit instrumental-variable regression by two-stage least squares (2SLS). This is equivalent to direct instrumental-variables estimation when the number of instruments is equal to the number of regressors. Alternative robust-regression estimators are also provided, based on M-estimation (2SM) and MM-estimation (2SMM).

## Usage

```
ivreg(
  formula,
  instruments,
  data,
  subset,
  na.action,
  weights,
  offset,
  contrasts = NULL,
  model = TRUE,
  y = TRUE,
  x = FALSE,
  method = c("OLS", "M", "MM"),
  ...
)
```

## Arguments

`formula`, `instruments`
: formula specification(s) of the regression relationship and the instruments. Either `instruments` is missing and `formula` has three parts as in `y ~ x1 + x2 | z1 + z2 + z3` (recommended) or `formula` is `y ~ x1 + x2` and `instruments` is a one-sided formula `~ z1 + z2 + z3` (only for backward compatibility).

`data`
: an optional data frame containing the variables in the model. By default the variables are taken from the environment of the `formula`.

`subset`
: an optional vector specifying a subset of observations to be used in fitting the model.

`na.action`
: a function that indicates what should happen when the data contain `NA`s. The default is set by the `na.action` option.

`weights`
: an optional vector of weights to be used in the fitting process.

`offset`
: an optional offset that can be used to specify an a priori known component to be included during fitting.

`contrasts`
: an optional list. See the `contrasts.arg` of `model.matrix.default`.

| | |
|---|---|
| model, x, y | logicals. If TRUE the corresponding components of the fit (the model frame, the model matrices, the response) are returned. These components are necessary for computing regression diagnostics. |
| method | the method used to fit the stage 1 and 2 regression: "OLS" for traditional 2SLS regression (the default), "M" for M-estimation, or "MM" for MM-estimation, with the latter two robust-regression methods implemented via the rlm function in the **MASS** package. |
| ... | further arguments passed to ivreg.fit. |

## Details

ivreg is the high-level interface to the work-horse function ivreg.fit. A set of standard methods (including print, summary, vcov, anova, predict, residuals, terms, model.matrix, bread, estfun) is available and described in ivregMethods. For methods related to regression diagnostics, see ivregDiagnostics.

Regressors and instruments for ivreg are most easily specified in a formula with two parts on the right-hand side, e.g., y ~ x1 + x2 | z1 + z2 + z3, where x1 and x2 are the explanatory variables and z1, z2, and z3 are the instrumental variables. Note that exogenous regressors have to be included as instruments for themselves.

For example, if there is one exogenous regressor ex and one endogenous regressor en with instrument in, the appropriate formula would be y ~ en + ex | in + ex. Alternatively, a formula with three parts on the right-hand side can also be used: y ~ ex | en | in. The latter is typically more convenient, if there is a large number of exogenous regressors.

Moreover, two further equivalent specification strategies are possible that are typically less convenient compared to the strategies above. One option is to use an update formula with a . in the second part of the formula is used: y ~ en + ex | . - en + in. Another option is to use a separate formula for the instruments (only for backward compatibility with earlier versions): formula = y ~ en + ex, instruments = ~ in + ex.

Internally, all specifications are converted to the version with two parts on the right-hand side.

For convenience, to facilitate model comparisons, it is also possible to specify a linear regression with a single right-hand side ivreg(y ~ x1 + x2, ...) which treats all regressors as exogenous. Thus, this yields simply the results from a standard linear model fitted via lm(y ~ x1 + x2, ...) or rlm(y ~ x1 + x2, ...), respectively.

## Value

ivreg returns an object of class "ivreg" that inherits from class "lm", with the following components:

| | |
|---|---|
| coefficients | parameter estimates, from the stage-2 regression. |
| residuals | vector of model residuals. |
| residuals1 | matrix of residuals from the stage-1 regression. |
| residuals2 | vector of residuals from the stage-2 regression. |
| fitted.values | vector of predicted means for the response. |
| weights | either the vector of weights used (if any) or NULL (if none). |

| | |
|---|---|
| offset | either the offset used (if any) or NULL (if none). |
| estfun | a matrix containing the empirical estimating functions. |
| n | number of observations. |
| nobs | number of observations with non-zero weights. |
| p | number of columns in the model matrix x of regressors. |
| q | number of columns in the instrumental variables model matrix z |
| rank | numeric rank of the model matrix for the stage-2 regression. |
| df.residual | residual degrees of freedom for fitted model. |
| cov.unscaled | unscaled covariance matrix for the coefficients. |
| sigma | residual standard deviation. |
| qr | QR decomposition for the stage-2 regression. |
| qr1 | QR decomposition for the stage-1 regression. |
| rank1 | numeric rank of the model matrix for the stage-1 regression. |
| coefficients1 | matrix of coefficients from the stage-1 regression. |
| df.residual1 | residual degrees of freedom for the stage-1 regression. |
| exogenous | columns of the "regressors" matrix that are exogenous. |
| endogenous | columns of the "regressors" matrix that are endogenous. |
| instruments | columns of the "instruments" matrix that are instruments for the endogenous variables. |
| method | the method used for the stage 1 and 2 regressions, one of "OLS", "M", or "MM". |
| rweights | a matrix of robustness weights with columns for each of the stage-1 regressions and for the stage-2 regression (in the last column) if the fitting method is "M" or "MM", NULL if the fitting method is "OLS". |
| hatvalues | a matrix of hatvalues. For method = "OLS", the matrix consists of two columns, for each of the stage-1 and stage-2 regression; for method = "M" or "MM", there is one column for *each* stage=1 regression and for the stage-2 regression. |
| df.residual | residual degrees of freedom for fitted model. |
| call | the original function call. |
| formula | the model formula. |
| na.action | function applied to missing values in the model fit. |
| terms | a list with elements "regressors" and "instruments" containing the terms objects for the respective components. |
| levels | levels of the categorical regressors. |
| contrasts | the contrasts used for categorical regressors. |
| model | the full model frame (if model = TRUE). |
| y | the response vector (if y = TRUE). |
| x | a list with elements "regressors", "instruments", "projected", containing the model matrices from the respective components (if x = TRUE). "projected" is the matrix of regressors projected on the image of the instruments. |

## References

Greene, W.H. (2003) *Econometric Analysis*, 5th ed., Upper Saddle River: Prentice Hall.

## See Also

ivreg.fit, ivregDiagnostics, ivregMethods, lm, lm.fit

## Examples

```
## data
data("CigaretteDemand", package = "ivreg")

## model
m <- ivreg(log(packs) ~ log(rprice) + log(rincome) | salestax + log(rincome),
  data = CigaretteDemand)
summary(m)
summary(m, vcov = sandwich::sandwich, df = Inf)

## ANOVA
m2 <- update(m, . ~ . - log(rincome) | . - log(rincome))
anova(m, m2)
car::Anova(m)

## same model specified by formula with three-part right-hand side
ivreg(log(packs) ~ log(rincome) | log(rprice) | salestax, data = CigaretteDemand)

# Robust 2SLS regression
data("Kmenta", package = "ivreg")
Kmenta1 <- Kmenta
Kmenta1[20, "Q"] <- 95 # corrupted data
deq <- ivreg(Q ~ P + D | D + F + A, data=Kmenta) # demand equation, uncorrupted data
deq1 <- ivreg(Q ~ P + D | D + F + A, data=Kmenta1) # standard 2SLS, corrupted data
deq2 <- ivreg(Q ~ P + D | D + F + A, data=Kmenta1, subset=-20) # standard 2SLS, removing bad case
deq3 <- ivreg(Q ~ P + D | D + F + A, data=Kmenta1, method="MM") # 2SLS MM estimation
car::compareCoefs(deq, deq1, deq2, deq3)
round(deq3$rweights, 2) # robustness weights
```

---

# ivreg.fit — *Fitting Instrumental-Variable Regressions by 2SLS, 2SM, or 2SMM Estimation*

---

## Description

Fit instrumental-variable regression by two-stage least squares (2SLS). This is equivalent to direct instrumental-variables estimation when the number of instruments is equal to the number of predictors. Alternative robust-regression estimation is also supported, based on M-estimation (2SM) or MM-estimation (2SMM).

## Usage

```
ivreg.fit(
  x,
  y,
  z,
  weights,
  offset,
  method = c("OLS", "M", "MM"),
  rlm.args = list(),
  ...
)
```

## Arguments

| | |
|---|---|
| x | regressor matrix. |
| y | vector for the response variable. |
| z | instruments matrix. |
| weights | an optional vector of weights to be used in the fitting process. |
| offset | an optional offset that can be used to specify an a priori known component to be included during fitting. |
| method | the method used to fit the stage 1 and 2 regression: "OLS" for traditional 2SLS regression (the default), "M" for M-estimation, or "MM" for MM-estimation, with the latter two robust-regression methods implemented via the rlm function in the **MASS** package. |
| rlm.args | a list of optional arguments to be passed to the rlm function in the **MASS** package if robust regression is used for the stage 1 and 2 regressions. |
| ... | further arguments passed to lm.fit or lm.wfit, respectively. |

## Details

ivreg is the high-level interface to the work-horse function ivreg.fit. ivreg.fit is essentially a convenience interface to lm.fit (or lm.wfit) for first projecting x onto the image of z, then running a regression of y on the projected x, and computing the residual standard deviation.

## Value

ivreg.fit returns an unclassed list with the following components:

| | |
|---|---|
| coefficients | parameter estimates, from the stage-2 regression. |
| residuals | vector of model residuals. |
| residuals1 | matrix of residuals from the stage-1 regression. |
| residuals2 | vector of residuals from the stage-2 regression. |
| fitted.values | vector of predicted means for the response. |
| weights | either the vector of weights used (if any) or NULL (if none). |
| offset | either the offset used (if any) or NULL (if none). |

| | |
|---|---|
| estfun | a matrix containing the empirical estimating functions. |
| n | number of observations. |
| nobs | number of observations with non-zero weights. |
| p | number of columns in the model matrix x of regressors. |
| q | number of columns in the instrumental variables model matrix z |
| rank | numeric rank of the model matrix for the stage-2 regression. |
| df.residual | residual degrees of freedom for fitted model. |
| cov.unscaled | unscaled covariance matrix for the coefficients. |
| sigma | residual standard error; when method is "M" or "MM", this is based on the MAD of the residuals (around 0) — see mad. |
| x | projection of x matrix onto span of z. |
| qr | QR decomposition for the stage-2 regression. |
| qr1 | QR decomposition for the stage-1 regression. |
| rank1 | numeric rank of the model matrix for the stage-1 regression. |
| coefficients1 | matrix of coefficients from the stage-1 regression. |
| df.residual1 | residual degrees of freedom for the stage-1 regression. |
| exogenous | columns of the "regressors" matrix that are exogenous. |
| endogenous | columns of the "regressors" matrix that are endogenous. |
| instruments | columns of the "instruments" matrix that are instruments for the endogenous variables. |
| method | the method used for the stage 1 and 2 regressions, one of "OLS", "M", or "MM". |
| rweights | a matrix of robustness weights with columns for each of the stage-1 regressions and for the stage-2 regression (in the last column) if the fitting method is "M" or "MM", NULL if the fitting method is "OLS". |
| hatvalues | a matrix of hatvalues. For method = "OLS", the matrix consists of two columns, for each of the stage-1 and stage-2 regression; for method = "M" or "MM", there is one column for *each* stage-1 regression and for the stage-2 regression. |

## See Also

ivreg, lm.fit, lm.wfit, rlm, mad

## Examples

```
## data
data("CigaretteDemand", package = "ivreg")

## high-level interface
m <- ivreg(log(packs) ~ log(rprice) + log(rincome) | salestax + log(rincome),
  data = CigaretteDemand)

## low-level interface
y <- m$y
x <- model.matrix(m, component = "regressors")
```

```
z <- model.matrix(m, component = "instruments")
ivreg.fit(x, y, z)$coefficients
```

---

Kmenta *Partly Artificial Data on the U.S. Economy*

---

## Description

These are partly contrived data from Kmenta (1986), constructed to illustrate estimation of a simultaneous-equation econometric model. The data are an annual time-series for the U.S. economy from 1922 to 1941. The values of the exogenous variables D, and F, and A are real, while those of the endogenous variables Q and P are simulated according to the linear simultaneous equation model fit in the examples.

## Usage

```
data("Kmenta", package = "ivreg")
```

## Format

A data frame with 20 rows and 5 columns.

**Q** food consumption per capita.

**P** ratio of food prices to general consumer prices.

**D** disposable income in constant dollars.

**F** ratio of preceding year's prices received by farmers to general consumer prices.

**A** time in years.

## Source

Kmenta, J. (1986) *Elements of Econometrics*, 2nd ed., Macmillan.

## See Also

ivreg.

## Examples

```
data("Kmenta", package = "ivreg")
deq <- ivreg(Q ~ P + D     | D + F + A, data = Kmenta) # demand equation
seq <- ivreg(Q ~ P + F + A | D + F + A, data = Kmenta) # supply equation
summary(deq, tests = TRUE)
summary(seq, tests = TRUE)
```

`SchoolingReturns` *U.S. Returns to Schooling Data*

## Description

Data from the U.S. National Longitudinal Survey of Young Men (NLSYM) in 1976 but using some variables dating back to earlier years.

## Usage

```
data("SchoolingReturns", package = "ivreg")
```

## Format

A data frame with 3010 rows and 22 columns.

**wage** Raw wages in 1976 (in cents per hour).

**education** Education in 1976 (in years).

**experience** Years of labor market experience, computed as `age - education - 6`.

**ethnicity** Factor indicating ethnicity. Is the individual African-American (`"afam"`) or not (`"other"`)?

**smsa** Factor. Does the individual reside in a SMSA (standard metropolitan statistical area) in 1976?

**south** Factor. Does the individual reside in the South in 1976?

**age** Age in 1976 (in years).

**nearcollege** Factor. Did the individual grow up near a 4-year college?

**nearcollege2** Factor. Did the individual grow up near a 2-year college?

**nearcollege4** Factor. Did the individual grow up near a 4-year public or private college?

**enrolled** Factor. Is the individual enrolled in college in 1976?

**married** factor. Is the individual married in 1976?

**education66** Education in 1966 (in years).

**smsa66** Factor. Does the individual reside in a SMSA in 1966?

**south66** Factor. Does the individual reside in the South in 1966?

**feducation** Father's educational attainment (in years). Imputed with average if missing.

**meducation** Mother's educational attainment (in years). Imputed with average if missing.

**fameducation** Ordered factor coding family education class (from 1 to 9).

**kww** Knowledge world of work (KWW) score.

**iq** Normed intelligence quotient (IQ) score

**parents14** Factor coding living with parents at age 14: both parents, single mother, step parent, other

**library14** Factor. Was there a library card in home at age 14?

## Details

Investigating the causal link of schooling on earnings in a classical model for wage determinants is problematic because it can be argued that schooling is endogenous. Hence, one possible strategy is to use an exogonous variable as an instrument for the years of education. In his well-known study, Card (1995) uses geographical proximity to a college when growing up as such an instrument, showing that this significantly increases both the years of education and the wage level obtained on the labor market. Using instrumental variables regression Card (1995) shows that the estimated returns to schooling are much higher than when simply using ordinary least squares.

The data are taken from the supplementary material for Verbeek (2004) and are based on the work of Card (1995). The U.S. National Longitudinal Survey of Young Men (NLSYM) began in 1966 and included 5525 men, then aged between 14 and 24. Card (1995) employs labor market information from the 1976 NLSYM interview which also included information about educational attainment. Out of the 3694 men still included in that wave of NLSYM, 3010 provided information on both wages and education yielding the subset of observations provided in `SchoolingReturns`.

The examples replicate the results from Verbeek (2004) who used the simplest specifications from Card (1995). Including further region or family background characteristics improves the model significantly but does not affect much the main coefficients of interest, namely that of years of education.

## Source

Supplementary material for Verbeek (2004).

## References

Card, D. (1995). Using Geographical Variation in College Proximity to Estimate the Return to Schooling. In: Christofides, L.N., Grant, E.K., and Swidinsky, R. (eds.), *Aspects of Labour Market Behaviour: Essays in Honour of John Vanderkamp*, University of Toronto Press, Toronto, 201-222.

Verbeek, M. (2004). *A Guide to Modern Econometrics*, 2nd ed. John Wiley.

## Examples

```
## load data
data("SchoolingReturns", package = "ivreg")

## Table 5.1 in Verbeek (2004) / Table 2(1) in Card (1995)
## Returns to education: 7.4%
m_ols <- lm(log(wage) ~ education + poly(experience, 2, raw = TRUE) + ethnicity + smsa + south,
  data = SchoolingReturns)
summary(m_ols)

## Table 5.2 in Verbeek (2004) / similar to Table 3(1) in Card (1995)
m_red <- lm(education ~ poly(age, 2, raw = TRUE) + ethnicity + smsa + south + nearcollege,
  data = SchoolingReturns)
summary(m_red)

## Table 5.3 in Verbeek (2004) / similar to Table 3(5) in Card (1995)
## Returns to education: 13.3%
m_iv <- ivreg(log(wage) ~ education + poly(experience, 2, raw = TRUE) + ethnicity + smsa + south |
```

```
  nearcollege + poly(age, 2, raw = TRUE) + ethnicity + smsa + south,
  data = SchoolingReturns)
summary(m_iv)
```

# Index