Package ‘ivreg’

September 3, 2020

Title Two-Stage Least-Squares Regression with Diagnostics

Version 0.5-0

Description Instrumental variable estimation for linear models by two-stage least-squares (2SLS) regression. 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, stats

Suggests AER, effects (>= 4.2.0), knitr, insight, parallel, rmarkdown, sandwich, testthat

Encoding UTF-8

LazyData true

VignetteBuilder knitr

BugReports https://github.com/john-d-fox/ivreg/issues/

URL https://john-d-fox.github.io/ivreg/

RoxygenNote 7.1.1

NeedsCompilation no

Author John Fox [aut, cre] (0000-0002-1196-8012),
Christian Kleiber [aut] (<https://orcid.org/0000-0002-6781-4733>),
Achim Zeileis [aut] (<https://orcid.org/0000-0003-0918-3766>)

Maintainer John Fox <jfox@mcmaster.ca>

Repository CRAN

Date/Publication 2020-09-03 06:42:24 UTC
Description


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

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


See Also

*CigarettesSW*.

Examples

```r
## 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)
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 + incomedef + incomedef + 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")
```
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 'influence.ivreg'
hatvalues(model, type = c("stage2", "both", "maximum"), ...)

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

```
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(x, ...)

## 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(x, ...)  

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

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

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

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

### 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 resid-

ual 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 (num-
influence.ivreg

ber 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 \texttt{lapply} with argu-
ments \texttt{function(X,FUN,...)}. The default is to use a loop unless the \texttt{ncores}
argument is specified (see below).

\texttt{ncores} Numeric, number of cores to be used in parallel computations. If set to an
integer the \texttt{applyfun} is set to use either \texttt{parLapply} (on Windows) or \texttt{mclapply}
(otherwise) with the desired number of cores.

... arguments to be passed down.

\texttt{ylab} The vertical axis label.

\texttt{distribution} “t” (the default) or “norm”.

\texttt{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 \texttt{Boot}.

\texttt{method} only “case” (case resampling) is supported: see \texttt{Boot}.

main Main title for the graph.

Value
In the case of \texttt{influence.ivreg}, an object of class “influence.ivreg” with the following com-
ponents:

\texttt{coefficients} the estimated regression coefficients
\texttt{model} the model matrix
\texttt{dfbeta} influence on coefficients
\texttt{sigma} deleted values of the residual standard deviation
\texttt{dffits} overall influence on the regression coefficients
\texttt{cookd} Cook’s distances
\texttt{hatvalues} hatvalues
\texttt{rstudent} Studentized residuals
\texttt{df.residual} residual degrees of freedom

In the case of other methods, such as \texttt{rstudent.ivreg} or \texttt{rstudent.influence.ivreg}, the corre-
sponding diagnostic statistics. Many other methods (e.g., \texttt{crPlot.ivreg}, \texttt{avPlot.ivreg}, \texttt{Effect.ivreg})
draw graphs.

See Also
\texttt{ivreg}, \texttt{avPlots}, \texttt{crPlots}, \texttt{predictorEffects}, \texttt{qqPlot}, \texttt{influencePlot}, \texttt{infIndexPlot}, \texttt{Boot},
\texttt{outlierTest}, \texttt{spreadLevelPlot}, \texttt{ncvTest}. 
Examples

```r
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*

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.

Usage

```r
ivreg(
  formula, instruments,
  data, subset, na.action,
  weights, offset, contrasts = NULL,
  model = TRUE, y = TRUE, x = FALSE,
  ...
)
```

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 | x3 + x4`, or `formula` has two parts as in `y ~ x1 + x2 | x3 + x4 + x5` if `instruments` is not missing. The right-hand-side `formula` specifies the relationship between the dependent variable `y` and the regressors, and the left-hand-side `formula` specifies the relationships between the endogenous regressors and the instruments.
  - `data`: data frame
  - `subset`: subset expression
  - `na.action`: function determining how missing values are to be treated
  - `weights`: case weights
  - `offset`: a one-sided formula specifying the linear components of the model to use as offset terms
  - `contrasts`: a list of contrasts to be used for some or all of the factors
  - `model`: logical; If `FALSE`, the model frame is not returned
  - `y`: logical; If `TRUE`, the model frame is returned
  - `x`: logical; If `TRUE`, the model frame is returned
  - `...`: further arguments passed to the formula parser

- The following arguments are ignored:
  - `x` is ignored if `model = TRUE`
  - `y` is ignored if `model = TRUE`
  - `model` is ignored if both `x` and `y` are `TRUE`

Examples

```r
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)
```

The `ivreg` function is used to fit instrumental-variable regression models using two-stage least squares (2SLS). It is often used in econometrics and other fields where endogeneity is a concern and direct estimation of the regression relationship is not possible. The syntax allows for specifying both the relationship between the dependent variable and the exogenous variables (`formula`), and the relationship between the endogenous regressors and the instruments (`instruments`). Additional arguments allow for controlling aspects of model estimation and diagnostics, such as weighting observations, handling missing data, and returning diagnostic plots and summary statistics.
**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 ~ ex + en | ex + in`. Equivalently, this can be specified as `y ~ ex + en | . -en + in`, i.e., by providing an update formula with a `.` in the second part of the formula. The latter is typically more convenient, if there is a large number of exogenous regressors.

**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.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
levels objects for the respective components.
contrasts levels of the 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

See Also
ivreg.fit, ivregDiagnostics, ivregMethods, lm, lm.fit

Examples
```r
## 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)

---

### ivreg.fit

*Fitting Instrumental-Variable Regressions by 2SLS*

**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.

**Usage**

`ivreg.fit(x, y, z, weights, offset, ...)`

**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.
- `...` 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.
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.

See Also

ivreg, lm.fit, lm.wfit

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
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


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)
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 exogenous 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


Examples

```r
## 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 |
vcov.ivreg

Methods for "ivreg" Objects

Description

Various methods for processing "ivreg" objects; for diagnostic methods, see \texttt{ivregDiagnostics}.

Usage

\texttt{vcov(object, ...)}

\texttt{bread.ivreg(x, ...)}

\texttt{estfun.ivreg(x, ...)}

\texttt{vcovHC.ivreg(x, ...)}

\texttt{terms(x, component = c("regressors", "instruments", "full"), ...)}

\texttt{model.matrix(object, component = c("regressors", "projected", "instruments"), ...)}

\texttt{predict(object, newdata, type = c("response", "terms"), na.action = na.pass, ...)}

\texttt{print(x, digits = max(3, getOption("digits") - 3), ...)}
## 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'
update(object, formula., ..., evaluate = TRUE)

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

Effect.ivreg(focal.predictors, mod, ...)

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

find_formula.ivreg(x, ...)

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

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

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

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

Arguments

object, object2, model, mod

An object of class "ivreg".

... arguments to pass down.

x An object of class "ivreg" or "summary.ivreg".

component For terms, "regressors", "instruments", or "full"; for model.matrix, "projected", "regressors", or "instruments"; for formula, "regressors", "instruments", or "complete".

ewndata Values of predictors for which to obtain predicted values.

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".

na.action na method to apply to predictor values for predictions; default is na.pass.

digits For printing.

cov. Optional coefficient covariance matrix, or a function to compute the covariance matrix, to use in computing the model summary.

df Optional residual degrees of freedom to use in computing model summary.

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.

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.

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.

hypothesis.matrix, rhs For formulating a linear hypothesis; see the documentation for linearHypothesis for details.

See Also

ivreg, ivreg.fit, ivregDiagnostics
Index

* datasets
  CigaretteDemand, 2
  Kmenta, 13
  SchoolingReturns, 14

* regression
  ivreg, 8
  ivreg.fit, 11

alias.ivreg (vcov.ivreg), 16
Anova.ivreg (vcov.ivreg), 16
anova.ivreg (vcov.ivreg), 16
avPlot.ivreg (influence.ivreg), 4
avPlots, 4, 7
avPlots.ivreg (influence.ivreg), 4

Boot, 7
Boot.ivreg (influence.ivreg), 4
bread.ivreg (vcov.ivreg), 16

ceresPlot.ivreg (influence.ivreg), 4
ceresPlots.ivreg (influence.ivreg), 4
CigaretteDemand, 2
CigarettesSW, 2, 3
cooks.distance.ivreg (influence.ivreg), 4
cooks.distance.ivreg (influence.ivreg), 4
crPlot.ivreg (influence.ivreg), 4
crPlots, 4, 7
crPlots.ivreg (influence.ivreg), 4
deviance.ivreg (influence.ivreg), 4
dfbeta.ivreg (influence.ivreg), 4
dfbeta.ivreg (influence.ivreg), 4
Eff ect, 18
Effect.ivreg (vcov.ivreg), 16
estfun.ivreg (vcov.ivreg), 16
find_formula.ivreg (vcov.ivreg), 16
form ula, 18
formula.ivreg (vcov.ivreg), 16
hatvalues.influence.ivreg (influence.ivreg), 4
hatvalues.ivreg (influence.ivreg), 4
infIndexPlot, 7
infIndexPlot.influence.ivreg (influence.ivreg), 4
infIndexPlot.ivreg (influence.ivreg), 4
influencePlot, 7
influencePlot.influence.ivreg (influence.ivreg), 4
influencePlot.ivreg (influence.ivreg), 4
ivreg, 7, 8, 11–13, 18
ivreg.fit, 9, 10, 11, 18
ivregDiagnostics, 9, 10, 16, 18
ivregDiagnostics (influence.ivreg), 4
ivregMethods, 9, 10
ivregMethods (vcov.ivreg), 16

Kmenta, 13
lapply, 7
linearHypothesis, 18
linearHypothesis.ivreg (vcov.ivreg), 16
lm, 10
lm.fit, 10–12
lm.wfit, 11, 12
mclapply, 7
mcPlot.ivreg (influence.ivreg), 4
mcPlots.ivreg (influence.ivreg), 4
model.matrix, 18
model.matrix.default, 9
model.matrix.influence.ivreg (influence.ivreg), 4
model.matrix.ivreg (vcov.ivreg), 16
model.matrix.ivreg_projected
  (vcov.ivreg), 16

na.pass, 18
ncvTest, 7
ncvTest.ivreg(influence.ivreg), 4

outlierTest, 7
outlierTest.ivreg(influence.ivreg), 4

parLapply, 7
plot.ivreg(influence.ivreg), 4
predict.ivreg(vcov.ivreg), 16
predictorEffects, 4, 7
print.ivreg(vcov.ivreg), 16
print.summary.ivreg(vcov.ivreg), 16

qqPlot, 4, 7
qqPlot.influence.ivreg
       (influence.ivreg), 4
qqPlot.ivreg(influence.ivreg), 4
qr.ivreg(vcov.ivreg), 16

residuals.ivreg(vcov.ivreg), 16
rstudent.influence.ivreg
       (influence.ivreg), 4
rstudent.ivreg(influence.ivreg), 4

SchoolingReturns, 14
spreadLevelPlot, 7
spreadLevelPlot.ivreg
       (influence.ivreg), 4
StockWatson2007, 2
summary.ivreg(vcov.ivreg), 16

terms, 18
terms.ivreg(vcov.ivreg), 16

update.ivreg(vcov.ivreg), 16

vcov.ivreg, 16
vcovHC.ivreg(vcov.ivreg), 16