ivregress — Single-equation instrumental-variables regression

Description

ivregress fits linear models where one or more of the regressors are endogenously determined. ivregress supports estimation via two-stage least squares (2SLS), limited-information maximum likelihood (LIML), and generalized method of moments (GMM).

Quick start

2SLS estimation of a linear regression of y1 on x1 and endogenous regressor y2 that is instrumented by z1

```
ivregress 2sls y1 x1 (y2 = z1)
```

Same as above, but with two endogenous regressors, y2 and y3 instrumented by z1 and z2

```
ivregress 2sls y1 x1 (y2 y3 = z1 z2)
```

Same as above, but absorbing indicator variables for the levels of cvar1 and cvar2

```
ivregress 2sls y1 x1 (y2 y3 = z1 z2), absorb(cvar1 cvar2)
```

With robust standard errors

```
ivregress 2sls y1 x1 (y2 y3 = z1 z2), vce(robust)
```

Report small-sample statistics

```
ivregress 2sls y1 x1 (y2 y3 = z1 z2), small
```

Use LIML estimation

```
ivregress liml y1 x1 (y2 y3 = z1 z2)
```

Use GMM estimation

```
ivregress gmm y1 x1 (y2 y3 = z1 z2)
```

Also specify a weight matrix that allows for correlation within clusters identified by cvar

```
ivregress gmm y1 x1 (y2 y3 = z1 z2), wmatrix(cluster cvar)
```

Menu

Statistics > Endogenous covariates > Linear regression with endogenous covariates

Syntax

ivregress *estimator* *depvar* [*varlist*$_1$] (*varlist*$_2$ = *varlist*$_{iv}$) [*if*] [*in*] [*weight*]
[, *options*]

varlist$_1$ is the list of exogenous variables.

varlist$_2$ is the list of endogenous variables.

varlist$_{iv}$ is the list of exogenous variables used with *varlist*$_1$ as instruments for *varlist*$_2$.

estimator	Description
`2sls`	two-stage least squares (2SLS)
`liml`	limited-information maximum likelihood (LIML)
`gmm`	generalized method of moments (GMM)

options	Description
Model	
`absorb(`*varlist*[, *method*]`)`[1]	specify categorical variables to be absorbed
† `dfabsorb`[1]	adjust degrees of freedom for collinearity among absorbed variables
`noconstant`[1]	suppress constant term
`hascons`	has user-supplied constant
GMM[2]	
`wmatrix(`*wmtype*`)`	*wmtype* may be `robust`, `cluster` *clustvar*, `hac` *hacspec*, or `unadjusted`
`center`	center moments in weight matrix computation
`igmm`	use iterative instead of two-step GMM estimator
`eps(#)`[3]	specify parameter convergence criterion; default is `eps(1e-6)`
`weps(#)`[3]	specify weight-matrix convergence criterion; default is `weps(1e-6)`
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `unadjusted`, `robust`, `cluster` *clustvar*, `bootstrap`, `jackknife`, or `hac` *hacspec*
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`first`	report first-stage regression
`small`	make degrees-of-freedom adjustments and report small-sample statistics
`noheader`	display only the coefficient table
`depname(`*depname*`)`	substitute dependent variable name
`eform(`*string*`)`	report exponentiated coefficients and use *string* to label them
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
† *optimization_options*[4]	control the optimization process; seldom used
`perfect`	do not check for collinearity between endogenous regressors and excluded instruments
`coeflegend`	display legend instead of statistics

[1]These options may be specified only with 2sls.

[2]These options may be specified only with gmm.

[3]These options may be specified only with igmm.

[4]These options may be specified only with igmm or 2sls and absorb().

†Ignored if only one absorbed variable is specified.

varlist$_1$, *varlist*$_2$, and *varlist*$_{iv}$ may contain factor variables; see [U] **11.4.3 Factor variables**.

depvar, *varlist*$_1$, *varlist*$_2$, and *varlist*$_{iv}$ may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

bayesboot, bootstrap, by, collect, fmm, jackknife, rolling, statsby, and svy are allowed; see [U] **11.1.10 Prefix commands**. For more details, see [FMM] **fmm: ivregress**.

Weights are not allowed with the bootstrap prefix; see [R] **bootstrap**.

aweights are not allowed with the jackknife prefix; see [R] **jackknife**.

absorb(), dfabsorb, hascons, vce(), noheader, depname(), and weights are not allowed with the svy prefix; see [SVY] **svy**.

fweights, iweights, and pweights are not allowed with vce(hac).

aweights, fweights, iweights, and pweights are allowed; see [U] **11.1.6 weight**.

perfect and coeflegend do not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

Model

absorb(*varlist*[, *method*]) specifies the categorical variables to be absorbed. The results are adjusted as if indicator variables for each level of each variable in *varlist* were included in the regression.

The absorption of categorical variables involves projecting the *depvar* and all variables in *varlist*$_1$, *varlist*$_2$, and *varlist*$_{iv}$ via an alternating projection method (APM) iterative algorithm. *method* specifies the APM and is one of halperin or cimmino.

halperin, the default, uses the product of the projection matrices.

cimmino uses the mean of the projection matrices.

The two methods typically perform similarly. See Stammann (2018) for details.

method is ignored if only one absorbed variable is specified.

absorb() may not be combined with vce(hac).

dfabsorb adjusts the degrees of freedom to account for collinearity among absorbed variables. The default degrees of freedom assumes that all absorbed variables are independent. This option is ignored if only one absorbed variable is specified in absorb().

noconstant; see [R] **Estimation options**.

hascons indicates that a user-defined constant or its equivalent is specified among the independent variables.

GMM

wmatrix(*wmtype*) specifies the type of weight matrix to be used in conjunction with the GMM estimator.

wmatrix(robust), the default, requests a weight matrix that is optimal when the error term is heteroskedastic.

wmatrix(cluster *clustvar*) requests a weight matrix that accounts for arbitrary correlation among observations within clusters identified by *clustvar*.

wmatrix(hac *hacspec*) requests a heteroskedasticity- and autocorrelation-consistent (HAC) weight matrix. The full syntax of *hacspec* is one of the following:

wmatrix(hac *kernel* [#]) requests a HAC weight matrix using the specified kernel (see below) with optional # lags. The bandwidth of a kernel is equal to # + 1. If # is not specified, a kernel with $N-2$ lags is used, where N is the sample size.

wmatrix(hac *kernel* opt [#]) requests a HAC weight matrix using the specified kernel (see below), and the lag order is selected using Newey and West's (1994) optimal lag-selection algorithm. # is an optional tuning parameter that affects the lag order selected; see the discussion in *Methods and formulas*.

kernel may be one of the following:

bartlett or nwest requests the Bartlett (Newey–West) kernel.

parzen or gallant requests the Parzen (Gallant 1987) kernel.

quadraticspectral or andrews requests the quadratic spectral (Andrews 1991) kernel.

wmatrix(unadjusted) requests a weight matrix that is suitable when the errors are homoskedastic. The GMM estimator with this weight matrix is equivalent to the 2SLS estimator.

center requests that the sample moments be centered (demeaned) when computing GMM weight matrices. By default, centering is not done.

igmm requests that the iterative GMM estimator be used instead of the default two-step GMM estimator. Convergence is declared when the relative change in the parameter vector from one iteration to the next is less than eps() or the relative change in the weight matrix is less than weps().

eps(#) specifies the convergence criterion for successive parameter estimates when the iterative GMM estimator is used. The default is eps(1e-6). Convergence is declared when the relative difference between successive parameter estimates is less than eps() and the relative difference between successive estimates of the weight matrix is less than weps().

weps(#) specifies the convergence criterion for successive estimates of the weight matrix when the iterative GMM estimator is used. The default is weps(1e-6). Convergence is declared when the relative difference between successive parameter estimates is less than eps() and the relative difference between successive estimates of the weight matrix is less than weps().

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster *clustvar*), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [R] ***vce_option***.

vce(unadjusted), the default for 2sls and liml, specifies that an unadjusted (nonrobust) VCE matrix be used. The default for gmm is based on the *wmtype* specified in the wmatrix() option; see wmatrix() above. If wmatrix() is specified with gmm but vce() is not, then *vcetype* is set equal to *wmtype*. To override this behavior and obtain an unadjusted (nonrobust) VCE matrix, specify vce(unadjusted).

vce(hac *hacspec*) specifies that a HAC covariance matrix be used. The syntax is identical to that for wmatrix(). vce(hac) may not be combined with absorb().

Reporting

`level(#)`; see [R] **Estimation options**.

`first` requests that the first-stage regression results be displayed.

`small` requests that the degrees-of-freedom adjustment $N/(N-k)$ be made to the variance–covariance matrix of parameters and that small-sample F and t statistics be reported, where N is the sample size and k is the number of parameters estimated. By default, no degrees-of-freedom adjustment is made, and Wald and z statistics are reported. Even with this option, no degrees-of-freedom adjustment is made to the weight matrix when the GMM estimator is used.

`noheader` suppresses the display of the summary statistics at the top of the output, displaying only the coefficient table.

`depname(`*depname*`)` is used only in programs and ado-files that use `ivregress` to fit models other than instrumental-variables regression. `depname()` may be specified only at estimation time. *depname* is recorded as the identity of the dependent variable, even though the estimates are calculated using *depvar*. This method affects the labeling of the output—not the results calculated—but could affect later calculations made by `predict`, where the residual would be calculated as deviations from *depname* rather than *depvar*. `depname()` is most typically used when *depvar* is a temporary variable (see [P] **macro**) used as a proxy for *depname*.

`eform(`*string*`)` is used only in programs and ado-files that use `ivregress` to fit models other than instrumental-variables regression. `eform()` specifies that the coefficient table be displayed in "exponentiated form", as defined in [R] **Maximize**, and that *string* be used to label the exponentiated coefficients in the table.

display_options: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*style*`)`, `cformat(%`*fmt*`)`, `pformat(%`*fmt*`)`, `sformat(%`*fmt*`)`, and `nolstretch`; see [R] **Estimation options**.

Optimization

optimization_options: `iterate(#)`, `[no]log`, and `tolerance(#)`. `iterate()` specifies the maximum number of iterations to perform in conjunction with the iterative GMM estimator. The default is the number set using `set maxiter`, which is 300 by default. `log`/`nolog` specifies whether to show the iteration log; see `set iterlog` in [R] ***set iter***. `tolerance()` is allowed only with `2sls` and the `absorb()` option and specifies the projection tolerance. These options are seldom used.

The following options are available with `ivregress` but are not shown in the dialog box:

`perfect` requests that `ivregress` not check for collinearity between the endogenous regressors and excluded instruments, allowing one to specify "perfect" instruments. This option cannot be used with the LIML estimator. This option may be required when using `ivregress` to implement other estimators.

`coeflegend`; see [R] **Estimation options**.

Remarks and examples

`ivregress` performs instrumental-variables regression and weighted instrumental-variables regression. For a general discussion of instrumental variables, see Baum (2006), Cameron and Trivedi (2005; 2022, chap. 7) Davidson and MacKinnon (1993), Greene (2018, chap. 8), and Wooldridge (2010, 2020). See Hall (2005) for a lucid presentation of GMM estimation. Angrist and Pischke (2009, chap. 4) offer a casual yet thorough introduction to instrumental-variables estimators, including their use in estimating treatment effects. Some of the earliest work on simultaneous systems can be found in Cowles Commission monographs—Koopmans and Marschak (1950) and Koopmans and Hood (1953)—with the first developments of 2SLS appearing in Theil (1953) and Basmann (1957). However, Stock and Watson (2019, 401–402) present an example of the method of instrumental variables that was first published in 1928 by Philip Wright.

The syntax for `ivregress` assumes that you want to fit one equation from a system of equations or an equation for which you do not want to specify the functional form for the remaining equations of the system. To fit a full system of equations, using either 2SLS equation-by-equation or three-stage least squares, see [R] **reg3**. An advantage of `ivregress` is that you can fit one equation of a multiple-equation system without specifying the functional form of the remaining equations.

Formally, the model fit by `ivregress` is

$$y_i = \mathbf{y}_i\boldsymbol{\beta}_1 + \mathbf{x}_{1i}\boldsymbol{\beta}_2 + u_i \tag{1}$$

$$\mathbf{y}_i = \mathbf{x}_{1i}\boldsymbol{\Pi}_1 + \mathbf{x}_{2i}\boldsymbol{\Pi}_2 + \mathbf{v}_i \tag{2}$$

Here y_i is the dependent variable for the ith observation, $\mathbf{y}_i$ represents the endogenous regressors (*varlist*$_2$ in the syntax diagram), $\mathbf{x}_{1i}$ represents the included exogenous regressors (*varlist*$_1$ in the syntax diagram), and $\mathbf{x}_{2i}$ represents the excluded exogenous regressors (*varlist*$_{\text{iv}}$ in the syntax diagram). $\mathbf{x}_{1i}$ and $\mathbf{x}_{2i}$ are collectively called the instruments. u_i and $\mathbf{v}_i$ are zero-mean error terms, and the correlations between u_i and the elements of $\mathbf{v}_i$ are presumably nonzero.

The rest of the discussion is presented under the following headings:

2SLS and LIML estimators
GMM estimator
Video example

2SLS and LIML estimators

The most common instrumental-variables estimator is 2SLS.

▷ Example 1: 2SLS estimator

We have state data from the 1980 census on the median dollar value of owner-occupied housing (`hsngval`) and the median monthly gross rent (`rent`). We want to model `rent` as a function of `hsngval` and the percentage of the population living in urban areas (`pcturban`):

$$\texttt{rent}_i = \beta_0 + \beta_1\texttt{hsngval}_i + \beta_2\texttt{pcturban}_i + u_i$$

where i indexes states and u_i is an error term.

Because random shocks that affect rental rates in a state probably also affect housing values, we treat `hsngval` as endogenous. We believe that the correlation between `hsngval` and u is not equal to zero. On the other hand, we have no reason to believe that the correlation between `pcturban` and u is nonzero, so we assume that `pcturban` is exogenous.

Because we are treating `hsngval` as an endogenous regressor, we must have one or more additional variables available that are correlated with `hsngval` but uncorrelated with u. Moreover, these excluded exogenous variables must not affect `rent` directly, because if they do then they should be included in the regression equation we specified above. In our dataset, we have a variable for family income (`faminc`) and for region of the country (`region`) that we believe are correlated with `hsngval` but not the error term. Together, `pcturban`, `faminc`, and factor variables `2.region`, `3.region`, and `4.region` constitute our set of instruments.

To fit the equation in Stata, we specify the dependent variable and the list of included exogenous variables. In parentheses, we specify the endogenous regressors, an equal sign, and the excluded exogenous variables. Only the additional exogenous variables must be specified to the right of the equal sign; the exogenous variables that appear in the regression equation are automatically included as instruments.

Here we fit our model with the 2SLS estimator:

```
. use https://www.stata-press.com/data/r19/hsng
(1980 Census housing data)

. ivregress 2sls rent pcturban (hsngval = faminc i.region)

Instrumental-variables 2SLS regression            Number of obs   =         50
                                                  Wald chi2(2)    =      90.76
                                                  Prob > chi2     =     0.0000
                                                  R-squared       =     0.5989
                                                  Root MSE        =     22.166

------------------------------------------------------------------------------
        rent | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
     hsngval |   .0022398   .0003284     6.82   0.000     .0015961    .0028836
    pcturban |    .081516   .2987652     0.27   0.785    -.504053     .667085
       _cons |   120.7065   15.22839     7.93   0.000     90.85942    150.5536
------------------------------------------------------------------------------
Endogenous: hsngval
Exogenous:  pcturban faminc 2.region 3.region 4.region
```

As we would expect, states with higher housing values have higher rental rates. The proportion of a state's population that is urban does not have a significant effect on rents.

◁

❑ Technical note

In a simultaneous-equations framework, we could write the model we just fit as

$$\begin{aligned} \texttt{hsngval}_i &= \pi_0 + \pi_1\texttt{faminc}_i + \pi_2\texttt{2.region}_i + \pi_3\texttt{3.region}_i + \pi_4\texttt{4.region}_i + v_i \\ \texttt{rent}_i &= \beta_0 + \beta_1\texttt{hsngval}_i + \beta_2\texttt{pcturban}_i + u_i \end{aligned}$$

which here happens to be recursive (triangular), because `hsngval` appears in the equation for `rent` but `rent` does not appear in the equation for `hsngval`. In general, however, systems of simultaneous equations are not recursive. Because this system is recursive, we could fit the two equations individually via OLS if we were willing to assume that u and v were independent. For a more detailed discussion of triangular systems, see Kmenta (1997, 719–720).

Historically, instrumental-variables estimation and systems of simultaneous equations were taught concurrently, and older textbooks describe instrumental-variables estimation solely in the context of simultaneous equations. However, in recent decades, the treatment of endogeneity and instrumental-variables estimation has taken on a much broader scope, while interest in the specification of complete systems of simultaneous equations has waned. Most recent textbooks, such as Cameron and Trivedi (2005), Davidson and MacKinnon (1993), and Wooldridge (2010, 2020), treat instrumental-variables estimation as an integral part of the modern economists' toolkit and introduce it long before shorter discussions on simultaneous equations.

❑

In addition to the 2SLS member of the κ-class estimators, `ivregress` implements the LIML estimator. Both theoretical and Monte Carlo exercises indicate that the LIML estimator may yield less bias and confidence intervals with better coverage rates than the 2SLS estimator. See Poi (2006) and Stock, Wright, and Yogo (2002) (and the papers cited therein) for Monte Carlo evidence.

▷ Example 2: LIML estimator

Here we refit our model with the LIML estimator:

```
. ivregress liml rent pcturban (hsngval = faminc i.region)
Instrumental-variables LIML regression              Number of obs   =         50
                                                    Wald chi2(2)    =      75.71
                                                    Prob > chi2     =     0.0000
                                                    R-squared       =     0.4901
                                                    Root MSE        =     24.992
```

rent	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
hsngval	.0026686	.0004173	6.39	0.000	.0018507	.0034865
pcturban	-.1827391	.3571132	-0.51	0.609	-.8826681	.5171899
_cons	117.6087	17.22625	6.83	0.000	83.84587	151.3715

```
Endogenous: hsngval
Exogenous:  pcturban faminc 2.region 3.region 4.region
```

These results are qualitatively similar to the 2SLS results, although the coefficient on hsngval is about 19% higher.

◁

GMM estimator

Since the celebrated paper of Hansen (1982), the GMM has been a popular method of estimation in economics and finance, and it lends itself well to instrumental-variables estimation. The basic principle is that we have some *moment* or *orthogonality* conditions of the form

$$E(\mathbf{z}_i u_i) = \mathbf{0} \tag{3}$$

From (1), we have $u_i = y_i - \mathbf{y}_i\boldsymbol{\beta}_1 - \mathbf{x}_{1i}\boldsymbol{\beta}_2$. What are the elements of the instrument vector $\mathbf{z}_i$? By assumption, $\mathbf{x}_{1i}$ is uncorrelated with u_i, as are the excluded exogenous variables $\mathbf{x}_{2i}$, and so we use $\mathbf{z}_i = [\mathbf{x}_{1i}\ \mathbf{x}_{2i}]$. The moment conditions are simply the mathematical representation of the assumption that the instruments are exogenous—that is, the instruments are orthogonal to (uncorrelated with) u_i.

If the number of elements in $\mathbf{z}_i$ is just equal to the number of unknown parameters, then we can apply the analogy principle to (3) and solve

$$\frac{1}{N}\sum_i \mathbf{z}_i u_i = \frac{1}{N}\sum_i \mathbf{z}_i \left(y_i - \mathbf{y}_i\boldsymbol{\beta}_1 - \mathbf{x}_{1i}\boldsymbol{\beta}_2\right) = \mathbf{0} \tag{4}$$

This equation is known as the method of moments estimator. Here, where the number of instruments equals the number of parameters, the method of moments estimator coincides with the 2SLS estimator, which also coincides with what has historically been called the indirect least-squares estimator (Judge et al. 1985, 595).

The "generalized" in GMM addresses the case in which the number of instruments (columns of $\mathbf{z}_i$) exceeds the number of parameters to be estimated. Here there is no unique solution to the population moment conditions defined in (3), so we cannot use (4). Instead, we define the objective function

$$Q(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2) = \left(\frac{1}{N}\sum_i \mathbf{z}_i u_i\right)' \mathbf{W} \left(\frac{1}{N}\sum_i \mathbf{z}_i u_i\right) \tag{5}$$

where $\mathbf{W}$ is a positive-definite matrix with the same number of rows and columns as the number of columns of $\mathbf{z}_i$. $\mathbf{W}$ is known as the weight matrix, and we specify its structure with the `wmatrix()` option. The GMM estimator of $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ minimizes $Q(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$; that is, the GMM estimator chooses $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ to make the moment conditions as close to zero as possible for a given $\mathbf{W}$. For a more general GMM estimator, see [R] **gmm**. `gmm` does not restrict you to fitting a single linear equation, though the syntax is more complex.

A well-known result is that if we define the matrix $\mathbf{S}_0$ to be the covariance of $\mathbf{z}_i u_i$ and set $\mathbf{W} = \mathbf{S}_0^{-1}$, then we obtain the optimal two-step GMM estimator, where by optimal estimator we mean the one that results in the smallest variance given the moment conditions defined in (3).

Suppose that the errors u_i are heteroskedastic but independent among observations. Then

$$\mathbf{S}_0 = E(\mathbf{z}_i u_i u_i \mathbf{z}_i') = E(u_i^2 \mathbf{z}_i \mathbf{z}_i')$$

and the sample analogue is

$$\widehat{\mathbf{S}} = \frac{1}{N}\sum_i \widehat{u}_i^2 \mathbf{z}_i \mathbf{z}_i' \tag{6}$$

To implement this estimator, we need estimates of the sample residuals $\widehat{u}_i$. `ivregress gmm` obtains the residuals by estimating $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ by 2SLS and then evaluates (6) and sets $\mathbf{W} = \widehat{\mathbf{S}}^{-1}$. Equation (6) is the same as the center term of the "sandwich" robust covariance matrix available from most Stata estimation commands through the `vce(robust)` option.

▷ Example 3: GMM estimator

Here we refit our model of rents by using the GMM estimator, allowing for heteroskedasticity in u_i:

```
. ivregress gmm rent pcturban (hsngval = faminc i.region), wmatrix(robust)
Instrumental-variables GMM regression                Number of obs   =        50
                                                     Wald chi2(2)    =    112.09
                                                     Prob > chi2     =    0.0000
                                                     R-squared       =    0.6616
GMM weight matrix: Robust                            Root MSE        =    20.358
```

rent	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
hsngval	.0014643	.0004473	3.27	0.001	.0005877	.002341
pcturban	.7615482	.2895105	2.63	0.009	.1941181	1.328978
_cons	112.1227	10.80234	10.38	0.000	90.95052	133.2949

```
Endogenous: hsngval
Exogenous:  pcturban faminc 2.region 3.region 4.region
```

Because we requested that a heteroskedasticity-consistent weight matrix be used during estimation but did not specify the `vce()` option, `ivregress` reported standard errors that are robust to heteroskedasticity. Had we specified `vce(unadjusted)`, we would have obtained standard errors that would be correct only if the weight matrix $\mathbf{W}$ does in fact converge to $\mathbf{S}_0^{-1}$.

◁

❑ Technical note

Many software packages that implement GMM estimation use the same heteroskedasticity-consistent weight matrix we used in the previous example to obtain the optimal two-step estimates but do not use a heteroskedasticity-consistent VCE, even though they may label the standard errors as being "robust". To replicate results obtained from other packages, you may have to use the `vce(unadjusted)` option. See *Methods and formulas* below for a discussion of robust covariance matrix estimation in the GMM framework.

❑

By changing our definition of $\mathbf{S}_0$, we can obtain GMM estimators suitable for use with other types of data that violate the assumption that the errors are independent and identically distributed. For example, you may have a dataset that consists of multiple observations for each person in a sample. The observations that correspond to the same person are likely to be correlated, and the estimation technique should account for that lack of independence. Say that in your dataset, people are identified by the variable `personid` and you type

```
. ivregress gmm ..., wmatrix(cluster personid)
```

Here `ivregress` estimates $\mathbf{S}_0$ as

$$\hat{\mathbf{S}} = \frac{1}{N} \sum_{c \in C} \mathbf{q}_c \mathbf{q}_c'$$

where C denotes the set of clusters and

$$\mathbf{q}_c = \sum_{i \in c_j} \hat{u}_i \mathbf{z}_i$$

where c_j denotes the jth cluster. This weight matrix accounts for the within-person correlation among observations, so the GMM estimator that uses this version of $\mathbf{S}_0$ will be more efficient than the estimator that ignores this correlation.

▷ Example 4: GMM estimator with clustering

We have data from the National Longitudinal Survey on young women's wages as reported in a series of interviews from 1968 through 1988, and we want to fit a model of wages as a function of each woman's age and age squared, job tenure, birth year, and level of education. We believe that random shocks that affect a woman's wage also affect her job tenure, so we treat tenure as endogenous. As additional instruments, we use her union status, number of weeks worked in the past year, and a dummy indicating whether she lives in a metropolitan area. Because we have several observations for each woman (corresponding to interviews done over several years), we want to control for clustering on each person.

```
. use https://www.stata-press.com/data/r19/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)

. ivregress gmm ln_wage age c.age#c.age birth_yr grade
> (tenure = union wks_work msp), wmatrix(cluster idcode)

Instrumental-variables GMM regression             Number of obs     =     18,625
                                                  Wald chi2(5)      =    1807.17
                                                  Prob > chi2       =     0.0000
                                                  Root MSE          =     .46951

GMM weight matrix: Cluster (idcode)
                             (Std. err. adjusted for 4,110 clusters in idcode)

             |               Robust
     ln_wage | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      tenure |    .099221   .0037764    26.27   0.000     .0918194    .1066227
         age |   .0171146   .0066895     2.56   0.011     .0040034    .0302259
             |
 c.age#c.age |  -.0005191    .000111    -4.68   0.000    -.0007366   -.0003016
             |
    birth_yr |  -.0085994   .0021932    -3.92   0.000     -.012898   -.0043008
       grade |    .071574   .0029938    23.91   0.000     .0657062    .0774417
       _cons |   .8575071   .1616274     5.31   0.000     .5407231    1.174291
------------------------------------------------------------------------------
Endogenous: tenure
Exogenous:  age c.age#c.age birth_yr grade union wks_work msp
```

Both job tenure and years of schooling have significant positive effects on wages.

◁

Time-series data are often plagued by serial correlation. In these cases, we can construct a weight matrix to account for the fact that the error in period t is probably correlated with the errors in periods $t-1$, $t-2$, etc. A HAC weight matrix can be used to account for both serial correlation and potential heteroskedasticity.

To request a HAC weight matrix, you specify the `wmatrix(hac` *kernel* [#| `opt`]`)` option. *kernel* specifies which of three kernels to use: `bartlett`, `parzen`, or `quadraticspectral`. *kernel* determines the amount of weight given to lagged values when computing the HAC matrix, and # denotes the maximum number of lags to use. Many texts refer to the bandwidth of the kernel instead of the number of lags; the bandwidth is equal to the number of lags plus one. If neither `opt` nor # is specified, then $N-2$ lags are used, where N is the sample size.

If you specify `wmatrix(hac` *kernel* `opt)`, then `ivregress` uses Newey and West's (1994) algorithm for automatically selecting the number of lags to use. Although the authors' Monte Carlo simulations do show that the procedure may result in size distortions of hypothesis tests, the procedure is still useful when little other information is available to help choose the number of lags.

For more on GMM estimation, see Baum (2006); Baum, Schaffer, and Stillman (2003, 2007); Cameron and Trivedi (2005); Davidson and MacKinnon (1993); Hayashi (2000); or Wooldridge (2010). See Newey and West (1987) and Wang and Wu (2012) for an introduction to HAC covariance matrix estimation.

Video example

Instrumental variables regression using Stata

Stored results

`ivregress` stores the following in `e()`:

Scalars

e(N)	number of observations
e(k_absorb)	total number of absorbed categories
e(mss)	model sum of squares
e(df_m)	model degrees of freedom
e(rss)	residual sum of squares
e(df_r)	residual degrees of freedom
e(r2)	R^2
e(r2_a)	adjusted R^2
e(F)	F statistic
e(rmse)	root mean squared error
e(N_clust)	number of clusters
e(chi2)	χ^2
e(kappa)	κ used in LIML estimator
e(J)	value of GMM objective function
e(wlagopt)	lags used in HAC weight matrix (if Newey–West algorithm used)
e(vcelagopt)	lags used in HAC VCE matrix (if Newey–West algorithm used)
e(hac_lag)	HAC lag
e(rank)	rank of `e(V)`
e(k_endog)	number of endogenous regressors (after factor-variable expansion)
e(iterations)	number of GMM iterations (`0` if not applicable)

Macros

e(cmd)	`ivregress`
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(endog)	names of endogenous variables
e(exog)	names of exogenous variables
e(absvar)	names of absorbed variables
e(apm)	alternating projection method
e(constant)	`noconstant` or `hasconstant` if specified
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(hac_kernel)	HAC kernel
e(vce)	*vcetype* specified in `vce()`
e(vcetype)	title used to label Std. err.
e(estimator)	`2sls`, `liml`, or `gmm`
e(exogr)	exogenous regressors
e(wmatrix)	*wmtype* specified in `wmatrix()`
e(moments)	`centered` if `center` specified
e(small)	`small` if small-sample statistics
e(properties)	`b V`
e(estat_cmd)	program used to implement `estat`
e(predict)	program used to implement `predict`
e(footnote)	program used to implement footnote display
e(marginsok)	predictions allowed by `margins`
e(marginsnotok)	predictions disallowed by `margins`
e(asbalanced)	factor variables `fvset` as `asbalanced`
e(asobserved)	factor variables `fvset` as `asobserved`

Matrices

e(b)	coefficient vector
e(W)	weight matrix used to compute GMM estimates
e(S)	moment covariance matrix used to compute GMM variance–covariance matrix
e(V)	variance–covariance matrix of the estimators

`e(V_modelbased)`	model-based variance
`e(kabsorb)`	number of levels for each absorbed variable
`e(dfabsorb)`	adjusted degrees of freedom for each absorbed variable
`e(ksingle)`	number of singletons for each absorbed variable
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Methods and formulas are presented under the following headings:

Notation
2SLS and LIML estimators
2SLS estimator with absorb() option GMM estimator

Notation

Items printed in lowercase and italicized (for example, x) are scalars. Items printed in lowercase and boldfaced (for example, $\mathbf{x}$) are vectors. Items printed in uppercase and boldfaced (for example, $\mathbf{X}$) are matrices.

The model is

$$\mathbf{y} = \mathbf{Y}\boldsymbol{\beta}_1 + \mathbf{X}_1\boldsymbol{\beta}_2 + \mathbf{u} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$$
$$\mathbf{Y} = \mathbf{X}_1\boldsymbol{\Pi}_1 + \mathbf{X}_2\boldsymbol{\Pi}_2 + \mathbf{V} = \mathbf{Z}\boldsymbol{\Pi} + \mathbf{V}$$

where $\mathbf{y}$ is an $N \times 1$ vector of the left-hand-side variable; N is the sample size; $\mathbf{Y}$ is an $N \times p$ matrix of p endogenous regressors; $\mathbf{X}_1$ is an $N \times k_1$ matrix of k_1 included exogenous regressors; $\mathbf{X}_2$ is an $N \times k_2$ matrix of k_2 excluded exogenous variables, $\mathbf{X} = [\mathbf{Y}\ \mathbf{X}_1]$, $\mathbf{Z} = [\mathbf{X}_1\ \mathbf{X}_2]$; $\mathbf{u}$ is an $N \times 1$ vector of errors; $\mathbf{V}$ is an $N \times p$ matrix of errors; $\boldsymbol{\beta} = [\boldsymbol{\beta}_1\ \boldsymbol{\beta}_2]$ is a $k = (p + k_1) \times 1$ vector of parameters; and $\boldsymbol{\Pi}$ is a $(k_1 + k_2) \times p$ vector of parameters. If a constant term is included in the model, then one column of $\mathbf{X}_1$ contains all ones.

Let $\mathbf{v}$ be a column vector of weights specified by the user. If no weights are specified, $\mathbf{v} = \mathbf{1}$. Let $\mathbf{w}$ be a column vector of normalized weights. If no weights are specified or if the user specified `fweights` or `iweights`, $\mathbf{w} = \mathbf{v}$; otherwise, $\mathbf{w} = \{\mathbf{v}/(\mathbf{1}'\mathbf{v})\}(\mathbf{1}'\mathbf{1})$. Let $\mathbf{D}$ denote the $N \times N$ matrix with $\mathbf{w}$ on the main diagonal and zeros elsewhere. If no weights are specified, $\mathbf{D}$ is the identity matrix.

The weighted number of observations n is defined as $\mathbf{1}'\mathbf{w}$. For `iweights`, this is truncated to an integer. The *sum of the weights* is $\mathbf{1}'\mathbf{v}$. Define $c = 1$ if there is a constant in the regression and zero otherwise.

The order condition for identification requires that $k_2 \geq p$: the number of excluded exogenous variables must be at least as great as the number of endogenous regressors.

In the following formulas, if weights are specified, $\mathbf{X}_1'\mathbf{X}_1$, $\mathbf{X}'\mathbf{X}$, $\mathbf{X}'\mathbf{y}$, $\mathbf{y}'\mathbf{y}$, $\mathbf{Z}'\mathbf{Z}$, $\mathbf{Z}'\mathbf{X}$, and $\mathbf{Z}'\mathbf{y}$ are replaced with $\mathbf{X}_1'\mathbf{D}\mathbf{X}_1$, $\mathbf{X}'\mathbf{D}\mathbf{X}$, $\mathbf{X}'\mathbf{D}\mathbf{y}$, $\mathbf{y}'\mathbf{D}\mathbf{y}$, $\mathbf{Z}'\mathbf{D}\mathbf{Z}$, $\mathbf{Z}'\mathbf{D}\mathbf{X}$, and $\mathbf{Z}'\mathbf{D}\mathbf{y}$, respectively. We suppress the $\mathbf{D}$ below to simplify the notation.

2SLS and LIML estimators

Define the κ-class estimator of $\boldsymbol{\beta}$ as

$$\mathbf{b} = \left\{\mathbf{X}'(\mathbf{I} - \kappa\mathbf{M_Z})\mathbf{X}\right\}^{-1}\mathbf{X}'(\mathbf{I} - \kappa\mathbf{M_Z})\mathbf{y}$$

where $\mathbf{M_Z} = \mathbf{I} - \mathbf{Z}(\mathbf{Z}'\mathbf{Z})^{-1}\mathbf{Z}'$. The 2SLS estimator results from setting $\kappa = 1$. The LIML estimator results from selecting κ to be the minimum eigenvalue of $(\mathbf{Y}'\mathbf{M_Z}\mathbf{Y})^{-1/2}\mathbf{Y}'\mathbf{M}_{\mathbf{X}_1}\mathbf{Y}(\mathbf{Y}'\mathbf{M_Z}\mathbf{Y})^{-1/2}$, where $\mathbf{M}_{\mathbf{X}_1} = \mathbf{I} - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'$.

The total sum of squares (TSS) equals $\mathbf{y}'\mathbf{y}$ if there is no intercept and $\mathbf{y}'\mathbf{y} - \left\{(\mathbf{1}'\mathbf{y})^2/n\right\}$ otherwise. The degrees of freedom is $n - c$. The error sum of squares (ESS) is defined as $\mathbf{y}'\mathbf{y} - 2\mathbf{b}\mathbf{X}'\mathbf{y} + \mathbf{b}'\mathbf{X}'\mathbf{X}\mathbf{b}$. The model sum of squares (MSS) equals TSS − ESS. The degrees of freedom is $k - c$.

The mean squared error, s^2, is defined as ESS$/(n-k)$ if `small` is specified and ESS$/n$ otherwise. The root mean squared error is s, its square root.

If $c = 1$ and `small` is not specified, a Wald statistic, W, of the joint significance of the $k-1$ parameters of $\boldsymbol{\beta}$ except the constant term is calculated; $W \sim \chi^2(k-1)$. If $c = 1$ and `small` is specified, then an F statistic is calculated as $F = W/(k-1)$; $F \sim F(k-1, n-k)$.

The R^2 is defined as $R^2 = 1 - \text{ESS}/\text{TSS}$.

The adjusted R^2 is $R^2_\text{a} = 1 - (1 - R^2)(n - c)/(n - k)$.

The unadjusted (default) variance estimate is $\text{Var}(\mathbf{b}) = s^2\left\{\mathbf{X}'(\mathbf{I} - \kappa\mathbf{M_Z})\mathbf{X}\right\}^{-1}$.

For a general discussion of robust variance estimates in regression, see *A general notation for the robust variance calculation* in [R] **regress**. `ivregress` uses the same definitions for terms discussed in *Robust calculation for regress* in its robust variance calculation, except for the following.

The vector of scores is given by

$$\mathbf{u}_j = (y_j - \mathbf{x}_j\mathbf{b})\widehat{\mathbf{x}}_j$$

where $\widehat{\mathbf{x}}_j' = \mathbf{P}\mathbf{z}_j{}'$ and $\mathbf{P} = (\mathbf{X}'\mathbf{Z})(\mathbf{Z}'\mathbf{Z})^{-1}$. When the formulas in [R] **regress** are applied, q_c is given by its regressionlike definition. If `small` is not specified, then $k = 0$ in the formulas given in [R] **regress**.

`ivregress 2sls` and `ivregress liml` also support estimation with survey data. For details on VCEs with survey data, see [SVY] **Variance estimation**.

2SLS estimator with absorb() option

When absorbed variables are specified, we must project the dependent variable, instruments, and endogenous variables onto the orthogonal complement of the column space of the absorbed indicator matrices. Using the same notation found in the Methods and formulas of [R] **areg**, we have m_k categorical levels for the kth absorbed variable, C_k, and an $N \times m_k$ indicator matrix $\mathbf{D}_k$. The orthonormal projection matrix for the kth variable is $\mathbf{P}_k = \mathbf{D}_k(\mathbf{D}_k'\mathbf{D}_k)^{-1}\mathbf{D}_k'$. Thus, the product $\bar{\mathbf{y}}_k = \mathbf{P}_k\mathbf{y}$ is the projection of the dependent variable onto the column space of $\mathbf{D}_k$. That is, $\bar{\mathbf{y}}_k$ is the $N \times 1$ vector containing the (repeated) means of y_i for each level of C_k in the order that these levels appear in the sample. The product $(\mathbf{I} - \mathbf{P}_k)\mathbf{y}$ is the vector of the demeaned dependent variable. The same projection (demeaning) is applied to the columns of matrices $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{Y}$. The Halperin or Cimmino iterative algorithm loops over the h absorbed variables computing projections as described in Methods and formulas of [R] **areg**.

See *Methods and formulas* in [R] **areg** for a description of the method used to adjust the degrees of freedom to account for collinearity among absorbed variables when the `dfabsorb` option is specified.

GMM estimator

We obtain an initial consistent estimate of $\boldsymbol{\beta}$ by using the 2SLS estimator; see above. Using this estimate of $\boldsymbol{\beta}$, we compute the weight matrix $\mathbf{W}$ and calculate the GMM estimator

$$\mathbf{b}_{\text{GMM}} = \left\{\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X}\right\}^{-1}\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{y}$$

The variance of $\mathbf{b}_{\text{GMM}}$ is

$$\text{Var}(\mathbf{b}_{\text{GMM}}) = n\left\{\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X}\right\}^{-1}\mathbf{X}'\mathbf{Z}\mathbf{W}\widehat{\mathbf{S}}\mathbf{W}\mathbf{Z}'\mathbf{X}\left\{\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X}\right\}^{-1}$$

$\text{Var}(\mathbf{b}_{\text{GMM}})$ is of the sandwich form $\mathbf{DMD}$; see [P] _robust. If the user specifies the `small` option, `ivregress` implements a small-sample adjustment by multiplying the VCE by $N/(N-k)$.

If `vce(unadjusted)` is specified, then we set $\widehat{\mathbf{S}} = \mathbf{W}^{-1}$ and the VCE reduces to the "optimal" GMM variance estimator

$$\text{Var}(\boldsymbol{\beta}_{\text{GMM}}) = n\left\{\mathbf{X}'\mathbf{Z}\mathbf{W}\mathbf{Z}'\mathbf{X}\right\}^{-1}$$

However, if $\mathbf{W}^{-1}$ is not a good estimator of $E(\mathbf{z}_i u_i u_i \mathbf{z}_i')$, then the optimal GMM estimator is inefficient, and inference based on the optimal variance estimator could be misleading.

$\mathbf{W}$ is calculated using the residuals from the initial 2SLS estimates, whereas $\mathbf{S}$ is estimated using the residuals based on $\mathbf{b}_{\text{GMM}}$. The `wmatrix()` option affects the form of $\mathbf{W}$, whereas the `vce()` option affects the form of $\mathbf{S}$. Except for different residuals being used, the formulas for $\mathbf{W}^{-1}$ and $\mathbf{S}$ are identical, so we focus on estimating $\mathbf{W}^{-1}$.

If `wmatrix(unadjusted)` is specified, then

$$\mathbf{W}^{-1} = \frac{s^2}{n}\sum_i \mathbf{z}_i\mathbf{z}_i'$$

where $s^2 = \sum_i u_i^2/n$. This weight matrix is appropriate if the errors are homoskedastic.

If `wmatrix(robust)` is specified, then

$$\mathbf{W}^{-1} = \frac{1}{n}\sum_i u_i^2\mathbf{z}_i\mathbf{z}_i'$$

which is appropriate if the errors are heteroskedastic.

If `wmatrix(cluster` *clustvar*`)` is specified, then

$$\mathbf{W}^{-1} = \frac{1}{n}\sum_c \mathbf{q}_c\mathbf{q}_c'$$

where c indexes clusters,

$$\mathbf{q}_c = \sum_{i\in c_j} u_i\mathbf{z}_i$$

and c_j denotes the jth cluster.

If `wmatrix(hac` *kernel* [#]`)` is specified, then

$$\mathbf{W}^{-1} = \frac{1}{n}\sum_i u_i^2\mathbf{z}_i\mathbf{z}_i' + \frac{1}{n}\sum_{l=1}^{l=n-1}\sum_{i=l+1}^{i=n} K(l,m)u_i u_{i-l}\left(\mathbf{z}_i\mathbf{z}_{i-l}' + \mathbf{z}_{i-l}\mathbf{z}_i'\right)$$

where $m = \#$ if $\#$ is specified and $m = n - 2$ otherwise. Define $z = l/(m+1)$. If *kernel* is `nwest`, then

$$K(l,m) = \begin{cases} 1-z & 0 \le z \le 1 \\ 0 & \text{otherwise} \end{cases}$$

If *kernel* is `gallant`, then

$$K(l,m) = \begin{cases} 1-6z^2+6z^3 & 0 \le z \le 0.5 \\ 2(1-z)^3 & 0.5 < z \le 1 \\ 0 & \text{otherwise} \end{cases}$$

If *kernel* is `quadraticspectral`, then

$$K(l,m) = \begin{cases} 1 & z = 0 \\ 3\left\{\sin(\theta)/\theta - \cos(\theta)\right\}/\theta^2 & \text{otherwise} \end{cases}$$

where $\theta = 6\pi z/5$.

If `wmatrix(hac` *kernel* `opt)` is specified, then `ivregress` uses Newey and West's (1994) automatic lag-selection algorithm, which proceeds as follows. Define $\mathbf{h}$ to be a $(k_1 + k_2) \times 1$ vector containing ones in all rows except for the row corresponding to the constant term (if present); that row contains a zero. Define

$$\begin{aligned} f_i &= (u_i \mathbf{z}_i)\mathbf{h} \\ \widehat{\sigma}_j &= \frac{1}{n}\sum_{i=j+1}^{n} f_i f_{i-j} \qquad j = 0, \ldots, m^* \\ \widehat{s}^{(q)} &= 2\sum_{j=1}^{m^*} \widehat{\sigma}_j j^q \\ \widehat{s}^{(0)} &= \widehat{\sigma}_0 + 2\sum_{j=1}^{m^*} \widehat{\sigma}_j \\ \widehat{\gamma} &= c_\gamma \left\{ \left(\frac{\widehat{s}^{(q)}}{\widehat{s}^{(0)}} \right)^2 \right\}^{1/2q+1} \\ m &= \widehat{\gamma} n^{1/(2q+1)} \end{aligned}$$

where q, m^*, and c_γ depend on the kernel specified:

Kernel	q	m^*	c_γ
Bartlett	1	$\text{int}\left\{20(T/100)^{2/9}\right\}$	1.1447
Parzen	2	$\text{int}\left\{20(T/100)^{4/25}\right\}$	2.6614
Quadratic spectral	2	$\text{int}\left\{20(T/100)^{2/25}\right\}$	1.3221

where $\text{int}(x)$ denotes the integer obtained by truncating x toward zero. For the Bartlett and Parzen kernels, the optimal lag is $\min\{\text{int}(m), m^*\}$. For the quadratic spectral, the optimal lag is $\min\{m, m^*\}$.

If `wmatrix(hac` *kernel* `opt #)` is specified, then `ivregress` uses # instead of 20 in the definition of m^* above to select the optimal lag.

If center is specified, when computing weight matrices ivregress replaces the term $u_i z_i$ in the formulas above with $u_i\mathbf{z}_i - \overline{u\mathbf{z}}$, where $\overline{u\mathbf{z}} = \sum_i u_i\mathbf{z}_i/N$.

References

Anatolyev, S., and A. Skolkova. 2019. Many instruments: Implementation in Stata. *Stata Journal* 19: 849–866.

Andrews, D. W. K. 1991. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* 59: 817–858. https://doi.org/10.2307/2938229.

Angrist, J. D., and J.-S. Pischke. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.

Basmann, R. L. 1957. A generalized classical method of linear estimation of coefficients in a structural equation. *Econometrica* 25: 77–83. https://doi.org/10.2307/1907743.

Bauldry, S. 2014. miivfind: A command for identifying model-implied instrumental variables for structural equation models in Stata. *Stata Journal* 14: 60–75.

Baum, C. F. 2006. *An Introduction to Modern Econometrics Using Stata*. College Station, TX: Stata Press.

Baum, C. F., and A. Lewbel. 2019. Advice on using heteroskedasticity-based identification. *Stata Journal* 19: 757–767.

Baum, C. F., M. E. Schaffer, and S. Stillman. 2003. Instrumental variables and GMM: Estimation and testing. *Stata Journal* 3: 1–31.

———. 2007. Enhanced routines for instrumental variables/generalized method of moments estimation and testing. *Stata Journal* 7: 465–506.

Baum, C. F., D. Tommasi, and L. Zhang. 2024. Estimating treatment effects when program participation is misreported. *Stata Journal* 24: 614–629.

Benson, D., M. A. Masten, and A. Torgovitsky. 2022. ivcrc: An instrumental-variables estimator for the correlated random-coefficients model. *Stata Journal* 22: 469–495.

Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.

———. 2022. *Microeconometrics Using Stata*. 2nd ed. College Station, TX: Stata Press.

Choi, J., and S. Shen. 2019. Two-sample instrumental-variables regression with potentially weak instruments. *Stata Journal* 19: 581–597.

Clarke, D., and B. Matta. 2018. Practical considerations for questionable IVs. *Stata Journal* 18: 663–691.

Davidson, R., and J. G. MacKinnon. 1993. *Estimation and Inference in Econometrics*. New York: Oxford University Press.

Deb, P., E. C. Norton, and W. G. Manning. 2017. *Health Econometrics Using Stata*. College Station, TX: Stata Press.

Desbordes, R., and V. Verardi. 2012. A robust instrumental-variables estimator. *Stata Journal* 12: 169–181.

D'Haultfœuille, X., A. Maurel, X. Qiu, and Y. Zhang. 2020. Estimating selection models without an instrument with Stata. *Stata Journal* 20: 297–308.

Dippel, C., A. Ferrara, and S. Heblich. 2020. Causal mediation analysis in instrumental-variables regressions. *Stata Journal* 20: 613–626.

Du, K., Y. Zhang, and Q. Zhou. 2020. Fitting partially linear functional-coefficient panel-data models with Stata. *Stata Journal* 20: 976–998.

Finlay, K., and L. M. Magnusson. 2009. Implementing weak-instrument robust tests for a general class of instrumental-variables models. *Stata Journal* 9: 398–421.

Gallant, A. R. 1987. *Nonlinear Statistical Models*. New York: Wiley. https://doi.org/10.1002/9780470316719.

Greene, W. H. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Hall, A. R. 2005. *Generalized Method of Moments*. Oxford: Oxford University Press.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica* 50: 1029–1054. https://doi.org/10.2307/1912775.

Hayashi, F. 2000. *Econometrics*. Princeton, NJ: Princeton University Press.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T.-C. Lee. 1985. *The Theory and Practice of Econometrics*. 2nd ed. New York: Wiley.

Kaplan, D. M. 2022. Smoothed instrumental variables quantile regression. *Stata Journal* 22: 379–403.

Kmenta, J. 1997. *Elements of Econometrics*. 2nd ed. Ann Arbor: University of Michigan Press. https://doi.org/10.3998/mpub.15701.

Koopmans, T. C., and W. C. Hood. 1953. *Studies in Econometric Method*. New York: Wiley.

Koopmans, T. C., and J. Marschak. 1950. *Statistical Inference in Dynamic Economic Models*. New York: Wiley.

Kripfganz, S., and V. Sarafidis. 2021. Instrumental-variable estimation of large-T panel-data models with common factors. *Stata Journal* 21: 659–686.

Lee, M., and C. Han. 2024. Ordinary least squares and instrumental-variables estimators for any outcome and heterogeneity. *Stata Journal* 24: 72–92.

Lee, S., S. Lee, J. Owusu, and Y. Shin. 2023. csa2sls: A complete subset approach for many instruments using Stata. *Stata Journal* 23: 932–941.

Lin, A., D. Tommasi, and L. Zhang. 2024. Bounding program benefits when participation is misreported: Estimation and inference with Stata. *Stata Journal* 24: 185–212.

Newey, W. K., and K. D. West. 1987. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica* 55: 703–708. https://doi.org/10.2307/1913610.

———. 1994. Automatic lag selection in covariance matrix estimation. *Review of Economic Studies* 61: 631–653. https://doi.org/10.2307/2297912.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Palmer, T. M., V. Didelez, R. R. Ramsahai, and N. A. Sheehan. 2011. Nonparametric bounds for the causal effect in a binary instrumental-variable model. *Stata Journal* 11: 345–367.

Pinzon, E. 2016. Estimation under omitted confounders, endogeneity, omitted variable bias, and related problems. *The Stata Blog: Not Elsewhere Classified*. https://blog.stata.com/2017/02/20/estimation-under-omitted-confounders-endogeneity-omitted-variable-bias-and-related-problems/.

Poi, B. P. 2006. Jackknife instrumental variables estimation in Stata. *Stata Journal* 6: 364–376.

Stammann, A. 2018. Fast and feasible estimation of generalized linear models with high-dimensional k-way fixed effects. arXiv:1707.01815 [stat.AP], https://doi.org/10.48550/arXiv.1707.01815.

Stock, J. H., and M. W. Watson. 2019. *Introduction to Econometrics*. 4th ed. New York: Pearson.

Stock, J. H., J. H. Wright, and M. Yogo. 2002. A survey of weak instruments and weak identification in generalized method of moments. *Journal of Business and Economic Statistics* 20: 518–529. https://doi.org/10.1198/073500102288618658.

Sun, L. 2018. Implementing valid two-step identification-robust confidence sets for linear instrumental-variables models. *Stata Journal* 18: 803–825.

Theil, H. 1953. *Repeated Least Squares Applied to Complete Equation Systems*. Mimeograph from the Central Planning Bureau, The Hague.

Wang, Q., and N. Wu. 2012. Long-run covariance and its applications in cointegration regression. *Stata Journal* 12: 515–542.

Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

———. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Wright, P. G. 1928. *The Tariff on Animal and Vegetable Oils*. New York: Macmillan.

Young, A. 2024. Nearly collinear robust procedures for 2SLS estimation. *Stata Journal* 24: 93–112.

Also see

[R] **ivregress postestimation** — Postestimation tools for ivregress

[R] **cfregress** — Control-function linear regression

[R] **gmm** — Generalized method of moments estimation

[R] **ivprobit** — Probit model with continuous endogenous covariates

[R] **ivqregress** — Instrumental-variables quantile regression

[R] **ivtobit** — Tobit model with continuous endogenous covariates

[R] **reg3** — Three-stage estimation for systems of simultaneous equations

[R] **regress** — Linear regression

[ERM] **eregress** — Extended linear regression

[FMM] **fmm: ivregress** — Finite mixtures of linear regression models with endogenous covariates

[SEM] **Intro 5** — Tour of models

[SP] **spivregress** — Spatial autoregressive models with endogenous covariates

[SVY] **svy estimation** — Estimation commands for survey data

[TS] **forecast** — Econometric model forecasting

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[U] 20 Estimation and postestimation commands

For suggested citations, see the FAQ on citing Stata documentation.