

[Description](#)[Remarks and examples](#)[Quick start](#)[Stored results](#)[Menu](#)[Methods and formulas](#)[Syntax](#)[Also see](#)[Options](#)

Description

`xpovregress` fits a lasso instrumental-variables linear regression model and reports coefficients along with standard errors, test statistics, and confidence intervals for specified covariates of interest. The covariates of interest may be endogenous or exogenous. The cross-fit partialing-out method is used to estimate effects for these variables and to select from potential control variables and instruments to be included in the model.

Quick start

Estimate a coefficient for endogenous `d1` in a linear regression of `y` on `d1`, and include `x1` to `x100` as potential control variables and `z1` to `z100` as potential instruments to be selected by lassos

```
xpovregress y (d1 = z1-z100), controls(x1-x100)
```

Same as above, and estimate the coefficient for the exogenous `d2`

```
xpovregress y d2 (d1 = z1-z100), controls(x1-x100)
```

Same as above, but use 20 folds instead of 10 for cross-fitting

```
xpovregress y d2 (d1 = z1-z100), controls(x1-x100) xfolds(20)
```

Same as above, but repeat the cross-fitting procedure 15 times, and average the results

```
xpovregress y d2 (d1 = z1-z100), controls(x1-x100) xfolds(20) ///
resample(15)
```

Use cross-validation (CV) instead of a plugin iterative formula to select the optimal λ^* in each lasso

```
xpovregress y d2 (d1 = z1-z100), controls(x1-x100) selection(cv)
```

Same as above, and set a random-number seed for reproducibility

```
xpovregress y d2 (d1 = z1-z100), controls(x1-x100) selection(cv) ///
rseed(28)
```

Specify CV for the lasso for `y` only, with the stopping rule criterion turned off

```
xpovregress y d2 (d1 = z1-z100), controls(x1-x100) ///
lasso(y, selection(cv), stop(0))
```

Same as above, but apply the option to the lassos for `y`, `d2`, and `d1`

```
xpovregress y d2 (d1 = z1-z100), controls(x1-x100) ///
lasso(*, selection(cv), stop(0))
```

Menu

Statistics > Lasso > Lasso inferential models > Continuous outcomes > Cross-fit partialing-out IV model

Syntax

```
xpovregress depvar [exovars] (endovars = instrumvars) [if] [in],
    controls([(alwaysvars)] othervars) [options]
```

Coefficients and standard errors are estimated for the exogenous variables, *exovars*, and the endogenous variables, *endovars*. The set of instrumental variables, *instrumvars*, may be high dimensional.

<i>options</i>	Description
Model	
* <u>controls</u> ([<i>(alwaysvars)</i>] <i>othervars</i>)	<i>alwaysvars</i> and <i>othervars</i> are control variables for <i>depvar</i> , <i>exovars</i> , and <i>endovars</i> ; <i>instrumvars</i> are an additional set of control variables that apply only to the <i>endovars</i> ; <i>alwaysvars</i> are always included; lassos choose whether to include or exclude <i>othervars</i>
<u>selection</u> (plugin)	use a plugin iterative formula to select an optimal value of the lasso penalty parameter λ^* for each lasso; the default
<u>selection</u> (cv)	use CV to select an optimal value of the lasso penalty parameter λ^* for each lasso
<u>selection</u> (adaptive)	use adaptive lasso to select an optimal value of the lasso penalty parameter λ^* for each lasso
<u>selection</u> (bic)	use BIC to select an optimal value of the lasso penalty parameter λ^* for each lasso
<u>sqrtlasso</u>	use square-root lassos
<u>xfolds</u> (#)	use # folds for cross-fitting
<u>resample</u> [(#)]	repeat sample splitting # times and average results
<u>technique</u> (dm11 dm12)	use either double machine learning 1 (dm11) or double machine learning 2 (dm12) estimation technique; dm12 is the default
<u>missingok</u>	after fitting lassos, ignore missing values in any <i>instrumvars</i> or <i>othervars</i> not selected, and include these observations in the final model
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>robust</u> (the default) or <u>cluster</u> <i>clustvar</i>
Reporting	
<u>level</u> (#)	set confidence level; default is level(95)
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
[<u>no</u>]log	display or suppress an iteration log
<u>verbose</u>	display a verbose iteration log
<u>rseed</u> (#)	set random-number seed
Advanced	
<u>lasso</u> (<i>varlist</i> , <i>lasso_options</i>)	specify options for the lassos for variables in <i>varlist</i> ; may be repeated
<u>sqrtlasso</u> (<i>varlist</i> , <i>lasso_options</i>)	specify options for square-root lassos for variables in <i>varlist</i> ; may be repeated

<code>reestimate</code>	refit the model after using <code>lassoselect</code> to select a different λ^*
<code>noheader</code>	do not display the header on the coefficient table
<code>coeflegend</code>	display legend instead of statistics

*`controls()` is required.

exovars, *endovars*, *instrumvars*, *alwaysvars*, and *othervars* may contain factor variables. Base levels of factor variables cannot be set for *instrumvars*, *alwaysvars*, and *othervars*. See [U] 11.4.3 Factor variables.

`collect` is allowed; see [U] 11.1.10 Prefix commands.

`reestimate`, `noheader`, and `coeflegend` do not appear in the dialog box.

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

Options

Model

`controls([(alwaysvars)] othervars)` specifies the set of control variables, which control for omitted variables. Control variables are also known as confounding variables. *alwaysvars* are variables that are always to be included in lassos. *alwaysvars* are optional. *othervars* are variables that lassos will choose to include or exclude. The instrumental variables, *instrumvars*, are an additional set of control variables, but they apply only to the *endovars*. `controls()` is required.

`xpovregress` fits lassos for *depvar* and each one of the *exovars* and *endovars*. The control variables for the lassos for *depvar* and *exovars* are *alwaysvars* (always included) and *othervars* (lasso will include or exclude). The control variables for lassos for *endovars* are *exovars* (always included), *alwaysvars* (always included), *instrumvars* (lasso will include or exclude), and *othervars* (lasso will include or exclude).

`selection(plugin | cv | adaptive | bic)` specifies the selection method for choosing an optimal value of the lasso penalty parameter λ^* for each lasso or square-root lasso estimation. Separate lassos are estimated for *depvar* and each variable in *varsofinterest*. Specifying `selection()` changes the selection method for all of these lassos. You can specify different selection methods for different lassos using the option `lasso()` or `sqrtlasso()`. When `lasso()` or `sqrtlasso()` is used to specify a different selection method for the lassos of some variables, they override the global setting made using `selection()` for the specified variables.

`selection(plugin)` is the default. It selects λ^* based on a “plugin” iterative formula dependent on the data. See [LASSO] lasso options.

`selection(cv)` selects the λ^* that gives the minimum of the CV function. See [LASSO] lasso options.

`selection(adaptive)` selects λ^* using the adaptive lasso selection method. It cannot be specified when `sqrtlasso` is specified. See [LASSO] lasso options.

`selection(bic)` selects the λ^* that gives the minimum of the BIC function. See [LASSO] lasso options.

`sqrtlasso` specifies that square-root lassos be done rather than regular lassos. The option `lasso()` can be used with `sqrtlasso` to specify that regular lasso be done for some variables, overriding the global `sqrtlasso` setting for these variables. See [LASSO] lasso options.

`xfolds(#)` specifies the number of folds for cross-fitting. The default is `xfolds(10)`.

`resample(#)` specifies that sample splitting be repeated and results averaged. This reduces the effects of the randomness of sample splitting on the estimated coefficients. Not specifying `resample` or `resample(#)` is equivalent to specifying `resample(1)`. In other words, by default no resampling is done. Specifying `resample` alone is equivalent to specifying `resample(10)`. That is, sample splitting is repeated 10 times. For each sample split, lassos are computed. So when this option is not specified, lassos are repeated `x folds(#)` times. But when `resample(#)` is specified, lassos are repeated `x folds(#) × resample(#)` times. Thus, while we recommend using `resample` to get final results, note that it can be an extremely time-consuming procedure.

`technique(dm11 | dm12)` specifies which cross-fitting technique is used, either double machine learning 1 (`dm11`) or double machine learning 2 (`dm12`). For both techniques, the initial estimation steps are the same. The sample is split into $K = \text{x folds}(\#)$ folds. Then, coefficients on the controls are estimated using only the observations not in the k th fold, for $k = 1, 2, \dots, K$. Moment conditions for the coefficients on the *varsofinterest* are formed using the observations in fold k . The default technique, `dm12`, solves the moment conditions jointly across all the observations. The optional technique, `dm11`, solves the moment conditions in each fold k to produce K different estimates, which are then averaged to form a single vector of estimates. See [Methods and formulas](#).

`missingok` specifies that, after fitting lassos, the estimation sample be redefined based on only the non-missing observations of variables in the final model. In all cases, any observation with missing values for *depvar*, *exovars*, *endovars*, *instrumvars*, *alwaysvars*, and *othervars* is omitted from the estimation sample for the lassos. By default, the same sample is used for calculation of the coefficients of the *exovars* and *endovars* and their standard errors.

When `missingok` is specified, the initial estimation sample is the same as the default, but the sample used for the calculation of the coefficients of the *exovars* and *endovars* can be larger. Now observations with missing values for any *instrumvars* and *othervars* not selected will be added to the estimation sample (provided there are no missing values for any of the variables in the final model).

`missingok` may produce more efficient estimates when data are missing completely at random. It does, however, have the consequence that estimation samples can change when selected variables differ in models fit using different selection methods. That is, when *instrumvars* and *othervars* contain missing values, the estimation sample for a model fit using the default `selection(plugin)` will likely differ from the estimation sample for a model fit using, for example, `selection(cv)`.

SE/Robust

`vce(vctype)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`) and that allow for intragroup correlation (`cluster clustvar`); see [\[R\] vce_option](#).

When `vce(cluster clustvar)` is specified, all lassos also account for clustering. For each lasso, this affects how the log-likelihood function is computed and how the sample is split in cross-validation; see [Methods and formulas](#) in [\[LASSO\] lasso](#). Specifying `vce(cluster clustvar)` may lead to different selected controls and therefore to different point estimates for your variable of interest when compared to the estimation that ignores clustering.

Reporting

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

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `no1stretch`; see [\[R\] Estimation options](#).

Optimization

`[no]` `log` displays or suppresses a log showing the progress of the estimation. By default, one-line messages indicating when each lasso estimation begins are shown. Specify `verbose` to see a more detailed log.

`verbose` displays a verbose log showing the iterations of each lasso estimation. This option is useful when doing `selection(cv)` or `selection(adaptive)`. It allows you to monitor the progress of the lasso estimations for these selection methods, which can be time consuming when there are many *othervars* specified in `controls()` or many *instrumvars*.

`rseed(#)` sets the random-number seed. This option can be used to reproduce results. `rseed(#)` is equivalent to typing `set seed #` prior to running `xpoivregress`. Random numbers are used to produce split samples for cross-fitting. So for all `selection()` options, if you want to reproduce your results, you must either use this option or use `set seed`. See [R] [set seed](#).

Advanced

`lasso(varlist, lasso_options)` lets you set different options for different lassos, or advanced options for all lassos. You specify a *varlist* followed by the options you want to apply to the lassos for these variables, where *varlist* consists of one or more variables from *depvar*, *exovars*, or *endovars*. `_all` or `*` may be used to specify *depvar* and all *exovars* and *endovars*. This option is repeatable as long as different variables are given in each specification. *lasso_options* are `selection(...)`, `grid(...)`, `stop(#)`, `tolerance(#)`, `dtolerance(#)`, and `cvtolerance(#)`. When `lasso(varlist, selection(...))` is specified, it overrides any global `selection()` option for the variables in *varlist*. It also overrides the global `sqrtlasso` option for these variables. See [LASSO] [lasso options](#).

`sqrtlasso(varlist, lasso_options)` works like the option `lasso()`, except square-root lassos for the variables in *varlist* are done rather than regular lassos. *varlist* consists of one or more variables from *depvar*, *exovars*, or *endovars*. This option is repeatable as long as different variables are given in each specification. *lasso_options* are `selection(...)`, `grid(...)`, `stop(#)`, `tolerance(#)`, `dtolerance(#)`, and `cvtolerance(#)`. When `sqrtlasso(varlist, selection(...))` is specified, it overrides any global `selection()` option for the variables in *varlist*. See [LASSO] [lasso options](#).

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

`reestimate` is an advanced option that refits the `xpoivregress` model based on changes made to the underlying lassos using `lassoselect`. After running `xpoivregress`, you can select a different λ^* for one or more of the lassos estimated by `xpoivregress`. After selecting λ^* , you type `xpoivregress, reestimate` to refit the `xpoivregress` model based on the newly selected λ^* .

`reestimate` may be combined only with reporting options.

`noheader` prevents the coefficient table header from being displayed.

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

Remarks and examples

`xpoivregress` performs cross-fit partialing-out lasso instrumental-variables linear regression. This command estimates coefficients, standard errors, and confidence intervals and performs tests for variables of interest, both exogenous and endogenous, while using lassos to select from among potential control variables and instruments.

The instrumental-variables linear regression model is

$$y = \mathbf{d}\alpha'_d + \mathbf{f}\alpha'_f + \mathbf{x}\beta' + \epsilon$$

where \mathbf{d} are the endogenous variables, \mathbf{f} are the exogenous variables for which we wish to make inferences, and \mathbf{x} are the potential control variables from which the lassos select. In addition, lassos select from potential instrumental variables, \mathbf{z} . `xpovregress` reports estimated coefficients for α_d and α_f . However, cross-fit partialing-out does not provide estimates of the coefficients on the control variables or their standard errors. No estimation results can be reported for β .

For an introduction to the cross-fit partialing-out lasso method for inference, as well as the double-selection and partialing-out methods, see [\[LASSO\] Lasso inference intro](#).

Examples that demonstrate how to use `xpovregress` and the other lasso inference commands are presented in [\[LASSO\] Inference examples](#). In particular, we recommend reading [1 Overview](#) for an introduction to the examples and to the `v1` command, which provides tools for working with the large lists of variables that are often included when using lasso methods. See [2 Fitting and interpreting inferential models](#) for comparisons of the different methods of fitting inferential models that are available in Stata. See [6 Fitting an inferential model with endogenous covariates](#) for examples and discussion specific to models that account for endogenous covariates.

If you are interested in digging deeper into the lassos that are used to select controls, see [5 Exploring inferential model lassos](#) in [\[LASSO\] Inference examples](#).

Stored results

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

Scalars

<code>e(N)</code>	number of observations
<code>e(N_clust)</code>	number of clusters
<code>e(k_varsofinterest)</code>	number of variables of interest
<code>e(k_controls)</code>	number of potential control variables
<code>e(k_controls_sel)</code>	number of selected control variables
<code>e(k_inst)</code>	number of potential instruments
<code>e(k_inst_sel)</code>	number of selected instruments
<code>e(df)</code>	degrees of freedom for test of variables of interest
<code>e(chi2)</code>	χ^2
<code>e(p)</code>	p -value for test of variables of interest
<code>e(n_xfolds)</code>	number of folds for cross-fitting
<code>e(n_resample)</code>	number of resamples
<code>e(rank)</code>	rank of $e(V)$

Macros

<code>e(cmd)</code>	<code>xpovregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(lasso_depvars)</code>	names of dependent variables for all lassos
<code>e(varsofinterest)</code>	variables of interest
<code>e(controls)</code>	potential control variables
<code>e(controls_sel)</code>	selected control variables
<code>e(exog)</code>	exogenous variables
<code>e(endog)</code>	endogenous variables
<code>e(inst)</code>	potential instruments
<code>e(inst_sel)</code>	selected instruments
<code>e(model)</code>	linear
<code>e(title)</code>	title in estimation output

e(clustvar)	name of cluster variable
e(chi2type)	Wald; type of χ^2 test
e(vce)	vctype specified in vce()
e(vctype)	title used to label Std. err.
e(rngstate)	random-number state used
e(properties)	b V
e(predict)	program used to implement predict
e(select_cmd)	program used to implement lassoselect
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(V)	variance–covariance matrix of the estimators

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

xpovregress implements cross-fit partialing-out lasso instrumental-variables regression. See *Methods and formulas* in [LASSO] **xporegress** for details about cross-fitting methods DML1 and DML2 and resampling the partitions. See *Methods and formulas* in [LASSO] **poivregress** for details about partialing-out lasso instrumental-variables regression. The model is

$$y = \mathbf{d}\alpha'_d + \mathbf{f}\alpha'_f + \mathbf{x}\beta' + \epsilon$$

where \mathbf{d} contains the J_d endogenous covariates of interest, \mathbf{f} contains the J_f exogenous covariates of interest, and \mathbf{x} contains the p_x controls. We also have p_z outside instrumental variables, denoted by \mathbf{z} , that are correlated with \mathbf{d} but not with ϵ . The number of controls in \mathbf{x} and the number of instruments in \mathbf{z} can be large and, in theory, can grow with the sample size; however, the number of nonzero elements in β and nonzero coefficients of \mathbf{z} must not be too large, which is to say that the model must be sparse. See *Stata commands for inference* in [LASSO] **Lasso intro** for a discussion on what it means for the model to be sparse.

Cross-fit partialing-out lasso instrumental-variables regression algorithm

1. Randomly partition the sample into K subsamples called folds.
2. Define I_k to be the observations in fold k , and define IC'_k to be the sample observations not in fold k .
3. For each $k = 1, \dots, K$, fill in the observations of $i \in I_k$ for the $J_d + J_f$ moment conditions that identify α . These moment conditions use out-of-sample estimates of the high-dimensional components estimated using the observations $i \in IC'_k$.

- a. Using the observations $i \in IC_k$, perform a linear lasso of y on \mathbf{x} to select controls $\tilde{\mathbf{x}}_{k,y}$.

This lasso can choose the lasso penalty parameter (λ^*) using one of the plugin estimators for a linear lasso, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.

- b. Using the observations $i \in IC_k$, fit a linear regression of y on $\tilde{\mathbf{x}}_{k,y}$, and let $\hat{\boldsymbol{\delta}}_k$ be the estimated coefficients on $\tilde{\mathbf{x}}_{k,y}$.
- c. For the observations $i \in I_k$, fill in the residual.

$$\tilde{\rho}_i = y_i - \tilde{\mathbf{x}}_{k,y,i} \hat{\boldsymbol{\delta}}_k'$$

- d. Using the observations $i \in IC_k$, for each $j = 1, \dots, J_d$, perform a linear lasso of d_j on \mathbf{f} , \mathbf{x} , and \mathbf{z} to select the controls $\tilde{\mathbf{x}}_{d,k,j}$ and the instruments $\tilde{\mathbf{z}}_{k,j}$.

Each of these lassos can choose the lasso penalty parameter (λ_j^*) using one of the plugin estimators for a linear lasso, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.

- e. Using the observations $i \in IC_k$, for each $j = 1, \dots, J_d$, fit a linear regression of d_j on \mathbf{f} , $\tilde{\mathbf{x}}_{d,k,j}$ and $\tilde{\mathbf{z}}_{k,j}$, and denote their coefficient estimates by $\hat{\boldsymbol{\pi}}_k$, $\hat{\boldsymbol{\gamma}}_{k,j}$, and $\hat{\boldsymbol{\theta}}_{k,j}$.
- f. For the observations $i \in I_k$, for each $j = 1, \dots, J_d$, fill in the prediction for $d_{j,i}$,

$$\hat{d}_{j,i} = \mathbf{f}_i \hat{\boldsymbol{\pi}}_k' + \tilde{\mathbf{x}}_{d,k,j,i} \hat{\boldsymbol{\gamma}}_{k,j}' + \tilde{\mathbf{z}}_{k,j,i} \hat{\boldsymbol{\theta}}_{k,j}'$$

- g. Using observations $i \in IC_k$, for each $j = 1, \dots, J_d$, perform a linear lasso of \hat{d}_j on \mathbf{x} , and let $\check{\mathbf{x}}_j$ be the selected controls.

Each of these lassos can choose the lasso penalty parameter (λ_j^*) using one of the plugin estimators for a linear lasso, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.

- h. Using observations $i \in IC_k$, for each $j = 1, \dots, J_d$, fit a linear regression of \hat{d}_j on $\check{\mathbf{x}}_j$, and let $\check{\boldsymbol{\gamma}}_j$ denote the coefficient estimates.
- i. For the observations $i \in I_k$, for each $j = 1, \dots, J_d$, fill in

$$\check{d}_{j,i} = \hat{d}_{j,i} - \check{\mathbf{x}}_{j,i} \check{\boldsymbol{\gamma}}_j'$$

- j. For the observations $i \in I_k$, for each $j = 1, \dots, J_d$, fill in

$$\tilde{d}_{j,i} = d_{j,i} - \check{\mathbf{x}}_{j,i} \check{\boldsymbol{\gamma}}_j'$$

- k. Using the observations $i \in IC_k$, for each $j = 1, \dots, J_f$, perform a linear lasso of f_j on \mathbf{x} to select the controls $\tilde{\mathbf{x}}_{f,k,j}$.

Each of these lassos can choose the lasso penalty parameter (λ_j^*) using one of the plugin estimators for a linear lasso, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.

- l. Using the observations $i \in IC_k$, for each $j = 1, \dots, J_f$, fit a linear regression of f_j on $\tilde{\mathbf{x}}_{f,k,j}$, and let $\hat{\boldsymbol{\gamma}}_{f,k,j}$ denote the coefficient estimates.

m. For the observations $i \in I_k$, for each $j = 1, \dots, J_f$, fill in the residuals

$$\tilde{f}_{j,i} = f_{j,i} - \tilde{\mathbf{x}}_{f,k,j} \tilde{\gamma}'_{f,k,j}$$

n. For the observations $i \in I_k$, form the vector of instruments

$$\mathbf{w}_i = (\tilde{d}_{1,i}, \dots, \tilde{d}_{J_d,i}, \tilde{f}_{1,i}, \dots, \tilde{f}_{J_f,i})$$

o. For the observations $i \in I_k$, form the vector of partialled-out covariates

$$\mathbf{p}_i = (\tilde{d}_{1,i}, \dots, \tilde{d}_{J_d,i}, \tilde{f}_{1,i}, \dots, \tilde{f}_{J_f,i})$$

4. Compute the point estimates.

For DML2, compute $\hat{\alpha}$ by solving the following sample-moment equations.

$$\frac{1}{n} \sum_{i=1}^n \mathbf{w}'_i (\tilde{\rho}_i - \mathbf{p}_i \alpha') = \mathbf{0}$$

For DML1, $\hat{\alpha}$ is given by

$$\hat{\alpha} = \frac{1}{K} \sum_{k=1}^K \hat{\alpha}_k$$

where $\hat{\alpha}_k$ is computed by solving the sample-moment equations

$$\frac{1}{n_k} \sum_{i \in I_k} \mathbf{w}'_i (\tilde{\rho}_i - \mathbf{p}_i \alpha'_k) = \mathbf{0}$$

and n_k is the number of observations in I_k .

5. The VCE is estimated by

$$\widehat{\text{Var}}(\hat{\alpha}) = \frac{1}{n} \hat{\mathbf{J}}_0^{-1} \hat{\Psi} (\hat{\mathbf{J}}_0^{-1})'$$

where

$$\hat{\Psi} = \frac{1}{K} \sum_{k=1}^K \hat{\Psi}_k$$

$$\hat{\Psi}_k = \frac{1}{n_k} \sum_{i \in I_k} \hat{\psi}_i \hat{\psi}'_i$$

$$\hat{\psi}_i = \mathbf{w}'_i (\tilde{\rho}_i - \mathbf{p}_i \hat{\alpha}')$$

$$\hat{\mathbf{J}}_0 = \frac{1}{K} \sum_{k=1}^K \left(\frac{1}{n_k} \sum_{i \in I_k} \hat{\psi}_i^a \right)$$

and

$$\hat{\psi}_i^a = \mathbf{w}'_i \mathbf{p}_i$$

See *Methods and formulas* in [LASSO] **lasso** for details on how the lassos in steps 3a, 3d, 3g, and 3k choose their penalty parameters (λ^*).

Also see

[LASSO] **lasso inference postestimation** — Postestimation tools for lasso inferential models

[LASSO] **povregress** — Partialing-out lasso instrumental-variables regression

[R] **ivregress** — Single-equation instrumental-variables regression

[U] **20 Estimation and postestimation commands**

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).