lassocoef — Display coefficients after lasso estimation results

Description

lassocoef displays a table showing the selected variables after one or more lasso estimation results. It can also display the values of the coefficient estimates. When used with stored results from two or more lassos, it can be used to view the overlap between sets of selected variables.

After ds, po, and xpo commands, and after telasso, lassocoef can be used to view coefficients for a single lasso or for multiple lassos displayed side by side.

Quick start

Display the selected variables after lasso, sqrtlasso, or elasticnet

```
lassocoef
```

Display the values of the postselection coefficients after lasso, sqrtlasso, or elasticnet

```
lassocoef, display(coef, postselection)
```

Display the penalized coefficients of the standardized variables after lasso, sqrtlasso, or elasticnet sorted by their absolute values in descending order

```
lassocoef, display(coef, standardized) sort(coef, standardized)
```

Compare which variables were selected from three different runs of lasso, where the estimation results are stored under the names mylasso1, mylasso2, and mylasso3

```
lassocoef mylasso1 mylasso2 mylasso3
```

Same as above, but display the penalized coefficients of the unstandardized variables sorted by the values of the penalized coefficients of the standardized variables

```
lassocoef mylasso1 mylasso2 mylasso3, display(coef, penalized) ///
   sort(coef, standardized)
```

After fitting a lasso logit model, display the exponentiated postselection coefficients, which are odds ratios, and specify their display format

```
lassocoef, display(coef, postselection eform format(%6.2f))
```

After any of the ds or po commands, display the selected variables in the lasso for the dependent variable y

```
lassocoef (., for(y))
```

Same as above, but display the penalized coefficients of the standardized variables in the lasso for y sorted by their absolute values

```
lassocoef (., for(y)), display(coef, standardized) ///
   sort(coef, standardized)
```

Same as above, but compare the lasso for y from the results stored in mydsregress with the lasso for y from the results stored in myporegress

```
lassocoef (mydsregress, for(y)) (myporegress, for(y)), ///
    display(coef, standardized) sort(coef, standardized)
```

After xpologit without resample, compare the variables selected by the lassos for x in each of the 10 cross-fit folds

```
lassocoef (myxpo, for(x) xfold(1)) ///
        (myxpo, for(x) xfold(2)) ///
        ⋮
        (myxpo, for(x) xfold(10))
```

After xpologit with resample, compare the variables selected by the lassos for x in each of the 10 cross-fit folds in the first resample

```
lassocoef (myxpo, for(x) xfold(1) resample(1)) ///
        (myxpo, for(x) xfold(2) resample(1)) ///
        ⋮
        (myxpo, for(x) xfold(10) resample(1))
```

After telasso, display the selected variables in the lasso for the outcome variable y at treatment levels 1 and 0

```
lassocoef (., for(y) tlevel(1)) (., for(y) tlevel(0))
```

Menu

Statistics > Postestimation

Syntax

For current estimation results

After lasso, sqrtlasso, *or* elasticnet

lassocoef [, *options*]

After ds *or* po

lassocoef (., for(*varspec*)) [, *options*]

After xpo *without* resample

lassocoef (., for(*varspec*) xfold(#)) [, *options*]

After xpo *with* resample

lassocoef (., for(*varspec*) xfold(#) resample(#)) [, *options*]

After telasso *for the outcome variable*

lassocoef (., for(*varspec*) tlevel(#)) [, *options*]

After telasso *for the treatment variable*

lassocoef (., for(*varspec*)) [, *options*]

After telasso *for the outcome variable with cross-fitting but without* resample

lassocoef (., for(*varspec*) tlevel(#) xfold(#)) [, *options*]

After telasso *for the treatment variable with cross-fitting but without* resample

lassocoef (., for(*varspec*) xfold(#)) [, *options*]

After telasso *for the outcome variable with cross-fitting and* resample

lassocoef (., for(*varspec*) tlevel(#) xfold(#) resample(#)) [, *options*]

After telasso *for the treatment variable with cross-fitting and* resample

lassocoef (., for(*varspec*) xfold(#) resample(#)) [, *options*]

For multiple stored estimation results

lassocoef [*estspec1* [*estspec2* ...]] [, *options*] *estspec* for lasso, sqrtlasso, and elasticnet is

name

estspec for ds and po models is

(*name*, for(*varspec*))

estspec for xpo without resample is

(*name*, for(*varspec*) xfold(#))

estspec for xpo with resample is

(*name*, for(*varspec*) xfold(#) resample(#))

estspec for the treatment model in telasso is

(*name*, for(*varspec*))

estspec for the outcome model at the treatment level # in telasso is

(*name*, for(*varspec*) tlevel(#))

estspec for the treatment model in telasso with cross-fitting but without resample is

(*name*, for(*varspec*) xfold(#))

estspec for the outcome model at the treatment level # in telasso with cross-fitting but without resample is

(*name*, for(*varspec*) tlevel(#) xfold(#))

estspec for the treatment model in telasso with resample is

(*name*, for(*varspec*) xfold(#) resample(#))

estspec for the outcome model at the treatment level # in telasso with resample is

(*name*, for(*varspec*) tlevel(#) xfold(#) resample(#))

name is the name of a stored estimation result. Either nothing or a period (.) can be used to specify the current estimation result. _all or * can be used to specify all stored estimation results when all stored results are lasso, sqrtlasso, or elasticnet.

varspec is *varname*, except after poivregress and xpoivregress, when it is either *varname* or pred(*varname*).

options	Description
Options	
display(x)	indicate selected variables with an x; the default
display(u)	same as display(x), except variables unavailable to be selected indicated with a u
display(coef [, *coef_di_opts*])	display coefficient values
sort(none)	order of variables as originally specified; the default
sort(names)	order by the names of the variables
sort(coef [, *coef_sort_opts*])	order by the absolute values of the coefficients in descending order
nofvlabel	display factor-variable level values rather than value labels
nolegend	report or suppress table legend
nolstretch	do not stretch the width of the table to accommodate long variable names

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

nofvlabel, nolegend, and nolstretch do not appear in the dialog box.

coef_di_opts	Description
standardized	display penalized coefficients of standardized variables; the default
penalized	display penalized coefficients of unstandardized variables
postselection	display postselection coefficients of unstandardized variables
eform	display $\exp(b)$ rather than the coefficient b
format(%*fmt*)	use numerical format %*fmt* for the coefficient values

coef_sort_opts	Description
standardized	sort by penalized coefficients of standardized variables
penalized	sort by penalized coefficients of unstandardized variables
postselection	sort by postselection coefficients of unstandardized variables

Options

Options

display(*displayspec*) specifies what to display in the table. The default is display(x).

Blank cells in the table indicate that the corresponding variable was not selected by the lasso or was not specified in the model.

For some variables without fitted values, a code that indicates the reason for omission is reported in the table.

Empty levels of factors and interactions are coded with the letter e.

Base levels of factors and interactions are coded with the letter b. Base levels can be set on *alwaysvars* (variables always included in the lasso) but not on *othervars* (the set of variables from which lasso selects).

Variables omitted because of collinearity are coded with the letter `o`. Lasso does not label as omitted any *othervars* because of collinearity. Collinear variables are simply not selected. Variables in *alwaysvars* can be omitted because of collinearity. See *Remarks and examples* in [LASSO] **Collinear covariates**.

`display(x)` displays an `x` in the table when the variable has been selected by the lasso; that is, it has a nonzero coefficient.

`display(u)` is the same as `display(x)`, except that when a variable was not specified in the model, `u` (for unavailable) is displayed instead of a blank cell.

`display(coef` [`, standardized penalized postselection eform format(`*%fmt*`)`] `)` specifies that coefficient values be displayed in the table.

`standardized` specifies that the penalized coefficients of the standardized variables be displayed. This is the default when `display(coef)` is specified without options. Penalized coefficients of the standardized variables are the coefficient values used in the estimation of the lasso penalty. See *Methods and formulas* in [LASSO] **lasso**.

`penalized` specifies that the penalized coefficients of the unstandardized variables be displayed. Penalized coefficients of the unstandardized variables are the penalized coefficients of the standardized variables with the standardization removed.

`postselection` specifies that the postselection coefficients of the unstandardized variables be displayed. Postselection coefficients of the unstandardized variables are obtained by fitting an ordinary model (`regress` for `lasso linear`, `logit` for `lasso logit`, `probit` for `lasso probit`, and `poisson` for `lasso poisson`) using the selected variables. See *Methods and formulas* in [LASSO] **lasso**.

`eform` displays coefficients in exponentiated form. For each coefficient, $\exp(b)$ rather than b is displayed. This option can be used to display odds ratios or incidence-rate ratios after the appropriate estimation command.

`format(`*%fmt*`)` specifies the display format for the coefficients in the table. The default is `format(%9.0g)`.

`sort(`*sortspec*`)` specifies that the rows of the table be ordered by specification given by *sortspec*.

`sort(none)` specifies that the rows of the table be ordered by the order the variables were specified in the model specification. This is the default.

`sort(names)` orders rows alphabetically by the variable names of the covariates. In the case of factor variables, main effects and nonfactor variables are displayed first in alphabetical order. Then, all two-way interactions are displayed in alphabetical order, then, all three-way interactions, and so on.

`sort(coef` [`, standardized penalized postselection`] `)` orders rows in descending order by the absolute values of the coefficients. When results from two or more estimation results are displayed, results are sorted first by the ordering for the first estimation result with rows representing coefficients not in the first estimation result last. Within the rows representing coefficients not in the first estimation result, the rows are sorted by the ordering for the second estimation result with rows representing coefficients not in the first or second estimation results last. And so on.

`standardized` orders rows in descending order by the absolute values of the penalized coefficients of the standardized variables. This is the default when `sort(coef)` is specified without options.

`penalized` orders rows in descending order by the absolute values of the penalized coefficients of the unstandardized variables.

`postselection` orders rows in descending order by the absolute values of the postselection coefficients of the unstandardized variables.

`nofvlabel` displays factor-variable level numerical values rather than attached value labels. This option overrides the `fvlabel` setting. See [R] **set showbaselevels**.

`nolegend` specifies that the legend at the bottom of the table not be displayed. By default, it is shown.

`nolstretch` specifies that the width of the table not be automatically widened to accommodate long variable names. When `nolstretch` is specified, names are abbreviated to make the table width no more than 79 characters. The default, `lstretch`, is to automatically widen the table up to the width of the Results window. To change the default, use `set lstretch off`.

Required options for *estspec* after `telasso`, `ds`, `po`, and `xpo`:

`for(`*varspec*`)` specifies a particular lasso after `telasso` or after a `ds`, `po`, or `xpo` estimation command fit using the option `selection(cv)`, `selection(adaptive)`, or `selection(bic)`. For all commands except `poivregress` and `xpoivregress`, *varspec* is always *varname*.

For the `ds`, `po`, and `xpo` commands except `poivregress` and `xpoivregress`, *varspec* is either *depvar*, the dependent variable, or one of *varsofinterest* for which inference is done.

For `poivregress` and `xpoivregress`, *varspec* is either *varname* or `pred(`*varname*`)`. The lasso for *depvar* is specified with its *varname*. Each of the endogenous variables have two lassos, specified by *varname* and `pred(`*varname*`)`. The exogenous variables of interest each have only one lasso, and it is specified by `pred(`*varname*`)`.

For `telasso`, *varspec* is either the outcome variable or the treatment variable.

This option is required after `telasso` and after the `ds`, `po`, and `xpo` commands.

`xfold(#)` specifies a particular lasso after an `xpo` estimation command or after `telasso` when the option `xfolds(#)` was specified. For each variable to be fit with a lasso, K lassos are done, one for each cross-fit fold, where K is the number of folds. This option specifies which fold, where $\# = 1, 2, \ldots, K$. `xfold(#)` is required after an `xpo` command and after `telasso` when the option `xfolds(#)` was specified.

`resample(#)` specifies a particular lasso after an `xpo` estimation command or after `telasso` fit using the option `resample(#)`. For each variable to be fit with a lasso, $R \times K$ lassos are done, where R is the number of resamples and K is the number of cross-fitting folds. This option specifies which resample, where $\# = 1, 2, \ldots, R$. `resample(#)`, along with `xfold(#)`, is required after an `xpo` command and after `telasso` with resampling.

`tlevel(#)` specifies the lasso for the outcome variable at the specified treatment level after `telasso`. This option is required to refer to the outcome model after `telasso`.

Remarks and examples

`lassocoef` lists the variables selected by a lasso and optionally lists the values of their coefficients. It is useful for comparing the results of multiple lassos. It shows how much overlap there is among the sets of selected variables from the lassos.

By default, `lassocoef` indicates only whether a variable was selected, marking a selected variable with an `x`. The option `display(coef,` *coef_type*`)` can be used to display the values of the coefficients.

Lassos store three different types of coefficients (*coef_types*). We refer to them as `standardized`, `penalized`, and `postselection`.

Before a lasso is fit, the potential variables in the model are standardized so that they each have mean 0 and standard deviation 1. `standardized` refers to the coefficients of the standardized variables exactly as estimated by the minimization of the objective function.

`penalized` refers to the coefficients from the minimization of the objective function with the standardization unwound. `standardized`, strictly speaking, gives the penalized coefficients of the standardized variables. `penalized` gives the penalized coefficients of the unstandardized variables.

`postselection` coefficients are computed by taking the selected variables and, for a linear lasso, estimating an ordinary least-squares linear regression with them, and using those coefficients. For a logit lasso, a logistic regression gives the postselection coefficients; for a probit lasso, a probit regression gives them; and for a Poisson lasso, a Poisson regression gives them.

`lassocoef` also has a `sort(coef,` *coef_type*`)` option, which controls the order in which the variables are listed. The most useful ordering is `sort(coef, standardized)`. It sorts the listing by the absolute values of the standardized coefficients with the largest displayed first. Variables with larger absolute values of their standardized coefficients take up a larger share of the lasso penalty, and so in this sense, they are "more important" for prediction than variables with smaller values.

▷ Example 1: lasso

We will show some uses of `lassocoef` after `lasso`.

Here is an example using `lasso` from [LASSO] **lasso examples**. We load the data and make the `vl` variable lists active.

```
. use https://www.stata-press.com/data/r19/fakesurvey_vl
(Fictitious survey data with vl)
. vl rebuild
Rebuilding vl macros ...
  (output omitted)
```

We fit the lasso.

```
. lasso linear q104 $idemographics $ifactors $vlcontinuous, rseed(1234)
10-fold cross-validation with 100 lambdas ...
Grid value 1:     lambda = .9090511   no. of nonzero coef. =   0
Folds: 1...5....10   CVF = 18.33331
  (output omitted)
Grid value 28:    lambda = .0737359   no. of nonzero coef. =  80
Folds: 1...5....10   CVF = 11.92887
... cross-validation complete ... minimum found
Lasso linear model                          No. of obs        =        914
                                            No. of covariates =        277
Selection: Cross-validation                 No. of CV folds   =         10
```

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	.9090511	0	-0.0010	18.33331
23	lambda before	.1174085	58	0.3543	11.82553
* 24	selected lambda	.1069782	64	0.3547	11.81814
25	lambda after	.0974746	66	0.3545	11.8222
28	last lambda	.0737359	80	0.3487	11.92887

```
* lambda selected by cross-validation.
```

By default, after only one lasso, `lassocoef` lists the variables selected by the lasso.

```
. lassocoef
```

	active
0.gender	x
0.q3	x
0.q4	x
0.q5	x
2.q6	x
0.q7	x
(output omitted)	
q111	x
q139	x
_cons	x

```
Legend:
  b - base level
  e - empty cell
  o - omitted
  x - estimated
```

`lassocoef` is intended to be used to compare multiple lassos. So let's store the results of this lasso before we fit another. See [LASSO] **estimates store** for more on storing and saving lasso results.

```
. estimates store lassocv
```

We fit an adaptive lasso.

```
. lasso linear q104 $idemographics $ifactors $vlcontinuous,
> selection(adaptive) rseed(1234)
  (output omitted)
Lasso linear model                          No. of obs         =        914
                                            No. of covariates  =        277
Selection: Adaptive                         No. of lasso steps =          2
Final adaptive step results
```

ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
29	first lambda	52.54847	0	-0.0011	18.3349
82	lambda before	.3794425	40	0.4077	10.84767
* 83	selected lambda	.3457338	41	0.4077	10.84764
84	lambda after	.3150198	42	0.4074	10.85301
128	last lambda	.0052548	62	0.3954	11.07398

```
* lambda selected by cross-validation in final adaptive step.
. estimates store lassoadaptive
```

Adaptive lasso selected 41 variables. Lasso selected 64. We can compare both the differences in selection and differences in the values of the coefficients. We use `lassocoef` with `display(coef, standardized)` to list the values of the standardized coefficients. We specify `sort(coef, standardized)` to sort them so that the largest ones in absolute value from the first lasso are shown first. The option `nofvlabel` means that numerical values for the factor-variable levels are displayed rather than value labels.

```
. lassocoef lassocv lassoadaptive, display(coef, standardized)
> sort(coef, standardized) nofvlabel nolegend
```

	lassocv	lassoadaptive
0.q19	-.8228234	-.9542076
0.q88	.7464342	.8650972
3.q156	-.6770033	-.770628
0.q48	-.6055556	-.7086328
0.q73	-.5962807	-.7036719
0.q85	-.5855315	-.684066
q31	.5843145	.7228376
0.q101	.5565875	.6682665

(output omitted)

0.q75	-.0056084	
q63	-.0055279	
0.q55	-.0054106	
0.q51	.0043129	
0.q77	.0019468	
0.q115	.0005097	
_cons	0	3.55e-15

Most of the differences occur in the coefficients with the smallest absolute values.

Let's fit another lasso. Note that we omitted the variable list `idemographics` from the potential variables this time.

```
. lasso linear q104 $ifactors $vlcontinuous, selection(cv) rseed(1234)
  (output omitted)
Lasso linear model                          No. of obs        =        916
                                            No. of covariates =        269
Selection: Cross-validation                 No. of CV folds   =         10

--------------------------------------------------------------------------
         |                                No. of      Out-of-      CV mean
         |                               nonzero       sample   prediction
      ID |     Description      lambda     coef.    R-squared        error
---------+----------------------------------------------------------------
       1 |    first lambda    .9127278         0      -0.0020     18.33925
      24 |   lambda before    .1074109        57       0.3406     12.06842
    * 25 | selected lambda    .0978688        62       0.3407     12.06704
      26 |    lambda after    .0891744        70       0.3400     12.07962
      28 |     last lambda    .0740342        78       0.3361     12.15082
--------------------------------------------------------------------------
* lambda selected by cross-validation.

. estimates store lassocv2
```

The option `display(u)` puts a u next to the variables that were unavailable to be selected.

```
. lassocoef lassocv lassocv2, display(u)

----------------------------------
             |  lassocv   lassocv2
-------------+--------------------
    0.gender |     x         u
        0.q3 |     x         u
        0.q4 |     x         u
        0.q5 |     x         u
             |
          q6 |
          2  |     x         x
          3  |               x
             |
  (output omitted)
        q100 |
          No |               x
         q21 |               x
         q52 |               x
       _cons |     x         x
----------------------------------
Legend:
  b - base level
  e - empty cell
  o - omitted
  x - estimated
  u - not selected for estimation
```

If `display(u)` was not specified, there would be empty space in place of the u's. So this option is useful for distinguishing whether a variable was not selected or simply not included in the model specification.

◁

▷ Example 2: poivregress

We want to show you some differences that arise when you fit models containing endogenous variables using poivregress and xpoivregress.

We will not describe the data or the model here. See [LASSO] **Inference examples**.

We load the data,

```
. use https://www.stata-press.com/data/r19/mroz2, clear
```

set vl variable lists,

```
. vl create vars       = (kidslt6 kidsge6 age husage city exper)
note: $vars initialized with 6 variables.
. vl substitute vars2 = c.vars c.vars#c.vars
. vl create iv         = (huseduc motheduc fatheduc)
note: $iv initialized with 3 variables.
. vl substitute iv2   = c.iv c.iv#c.iv
```

and fit our model using poivregress.

```
. poivregress lwage (educ = $iv2), controls($vars2) selection(cv) rseed(12345)
Estimating lasso for lwage using cv
Estimating lasso for educ using cv
Estimating lasso for pred(educ) using cv
Partialing-out IV linear model     Number of obs                  =        428
                                   Number of controls             =         27
                                   Number of instruments          =          9
                                   Number of selected controls    =         16
                                   Number of selected instruments =          4
                                   Wald chi2(1)                   =      11.10
                                   Prob > chi2                    =     0.0009
```

lwage	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
educ	.0765154	.0229707	3.33	0.001	.0314936	.1215371

```
Endogenous: educ
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.
. estimates store poivregresscv
```

We use lassoinfo to see the lassos fit by poivregress.

```
. lassoinfo poivregresscv
    Estimate: poivregresscv
     Command: poivregress
```

Variable	Model	Selection method	Selection criterion	lambda	No. of selected variables
lwage	linear	cv	CV min.	.0353704	3
educ	linear	cv	CV min.	.0530428	10
pred(educ)	linear	cv	CV min.	.013186	12

We have two lassos for educ, the endogenous variable in the model. One is named educ and the other pred(educ). To compare the coefficient estimates for these two lassos, we type

```
. lassocoef (poivregresscv, for(educ)) (poivregresscv, for(pred(educ))),
> display(coef, standardized) sort(coef, standardized) nolegend
```

	poivregresscv educ	poivregresscv pred(educ)
c.huseduc#c.huseduc	1.047956	
c.motheduc#c.fatheduc	.5574474	
c.kidsge6#c.kidsge6	-.2293016	-.274782
c.kidslt6#c.kidslt6	.1175937	
c.kidsge6#c.exper	.1087689	.2928483
c.motheduc#c.motheduc	.0813009	
c.huseduc#c.fatheduc	.0411326	
c.city#c.exper	.0207999	.1020498
c.husage#c.exper	.0077213	
c.kidsge6#c.city	-.0017114	
kidslt6		.5342914
c.kidslt6#c.kidsge6		-.2364133
kidsge6		-.2129479
husage		-.2091804
c.husage#c.city		.1396385
c.exper#c.exper		-.133589
c.kidslt6#c.exper		-.1322304
c.city#c.city		.1320515
c.kidslt6#c.city		.0237243
_cons	0	1.78e-15

◁

▷ Example 3: xporegress

The xpo commands fit many lassos. For each lasso fit by a po command, the corresponding xpo command fits xfolds(#) × resample(#) lassos. Cross-fitting randomly creates different divisions of the data for each resample. We expect that lasso will select different variables for different cross-fit folds and resamples. See [LASSO] **Inference examples** for a description of the data and model.

We load the data, set vl variable lists, fit our model using xporegress with the options xfolds(3) and resample(2), and then store the results with estimates store.

```
. use https://www.stata-press.com/data/r19/breathe, clear
(Nitrogen dioxide and attention)

. vl set
  (output omitted)

. vl move (siblings_old siblings_young) vlcontinuous
note: 2 variables specified and 2 variables moved.
  (output omitted)

. vl create mycontinuous   = vlcontinuous - (react no2_class)
note: $mycontinuous initialized with 10 variables.

. vl substitute mycontrols = i.vlcategorical mycontinuous

. xporegress react no2_class, controls($mycontrols) xfolds(3) resample(2)
> selection(cv) rseed(12345)

Resample 1 of 2 ...
Cross-fit fold 1 of 3 ...
Estimating lassos: 1.

Resample 1 of 2 ...
Cross-fit fold 2 of 3 ...
Estimating lassos: 1.

Resample 1 of 2 ...
Cross-fit fold 3 of 3 ...
Estimating lassos: 1.

Resample 2 of 2 ...
Cross-fit fold 1 of 3 ...
Estimating lassos: 1.

Resample 2 of 2 ...
Cross-fit fold 2 of 3 ...
Estimating lassos: 1.

Resample 2 of 2 ...
Cross-fit fold 3 of 3 ...
Estimating lassos: 1.

Cross-fit partialing-out            Number of obs               =      1,036
linear model                        Number of controls          =         32
                                    Number of selected controls =         27
                                    Number of folds in cross-fit =         3
                                    Number of resamples         =          2
                                    Wald chi2(1)                =      20.99
                                    Prob > chi2                 =     0.0000
```

react	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
no2_class	2.332193	.5090902	4.58	0.000	1.334394	3.329991

```
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.

. estimates store xpocv
```

For each cross-fit fold and each resample, `xporegress` fits lassos. It fit six lassos for the dependent variable, `react`, and six for the variable of interest, `no2_class`. To see how the variables selected differ for different folds and for different resamples, we type

```
. lassocoef (xpocv, for(react) resample(1) xfold(1))
>           (xpocv, for(react) resample(1) xfold(2))
>           (xpocv, for(react) resample(1) xfold(3))
>           (xpocv, for(react) resample(2) xfold(1))
>           (xpocv, for(react) resample(2) xfold(2))
>           (xpocv, for(react) resample(2) xfold(3))
>           , sort(coef, standardized)
```

	xpocv react_1_1	xpocv react_2_1	xpocv react_3_1	xpocv react_1_2	xpocv react_2_2	xpocv react_3_2
grade						
2nd	x	x	x	x	x	x
sex						
Male	x	x	x	x	x	x
grade						
4th	x	x	x	x	x	x
age	x	x	x	x	x	x
feducation						
University	x	x	x	x	x	x
age0	x	x	x		x	x
meducation						
Primary	x	x	x	x	x	x
breastfeed						
2	x	x			x	
0.msmoke	x	x			x	
feducation						
Primary		x	x	x		x
<Primary		x			x	
sev_school		x				x
meducation						
<Primary		x	x		x	x
(output omitted)						
0.overweight					x	
precip						x
green_home						x

```
Legend:
  b - base level
  e - empty cell
  o - omitted
  x - estimated
```

Even though we had `lassocoef` display x's, we specified the `sort(coef, standardized)` option so that the table is ordered by the most important variables from the lasso in the first column.

◁

Stored results

`lassocoef` stores the following in `r()`:

Macros

`r(names)`	names of results used

Matrices

`r(coef)`	matrix M: $n \times m$
	$M[i,\ j]$ = ith coefficient estimate for model j; $i = 1, \ldots, n$; $j = 1, \ldots, m$

Also see

[LASSO] **lasso postestimation** — Postestimation tools for lasso for prediction

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

[LASSO] **lassoinfo** — Display information about lasso estimation results

[CAUSAL] **telasso postestimation** — Postestimation tools for telasso

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