Title

stata.com

lassoselect — Select lambda after lasso

Description

`lassoselect` allows the user to select a different λ^* after `lasso` and `sqrtlasso` when the selection method was `selection(cv)`, `selection(adaptive)`, `selection(bic)`, or `selection(none)`.

After `elasticnet`, the user can select a different (α^*, λ^*) pair.

When the `telasso`, `ds`, `po`, and `xpo` commands fit models using `selection(cv)`, `selection(adaptive)`, or `selection(bic)` ([LASSO] **lasso options**), `lassoselect` can be used to select a different λ^* for a particular lasso.

Quick start

After `lasso` with `selection(cv)`, change the selected λ^* to that with ID $= 52$

```
lassoselect id = 52
```

Same as above, but change the selected λ^* to the λ closest to 0.01

```
lassoselect lambda = 0.01
```

After `elasticnet`, change the selected (α^*, λ^*) to $(0.5, 0.267345)$

```
lassoselect alpha = 0.5 lambda = 0.267345
```

After `dsregress` with `selection(adaptive)`, change the selected λ^* to 1.65278 for the adaptive lasso for the variable `y`

```
lassoselect lambda = 1.65278, for(y)
```

After `poivregress` with `selection(bic)`, change the selected λ^* to the λ closest to 0.7 for the lasso for the prediction of the variable `income`

```
lassoselect lambda = 0.7, for(pred(income))
```

After `xporegress` with `selection(cv)` and `resample`, change the selected λ^* to 0.234189 for the lasso for the variable `x26` for the 5th cross-fit fold in the 9th resample

```
lassoselect lambda = 0.234189, for(x26) xfold(5) resample(9)
```

After `telasso` with `selection(cv)`, change the selected λ^* to the λ closest to 0.7 for the lasso for the outcome variable `y` at treatment level 1

```
lassoselect lambda = 0.7, for(y) tlevel(1)
```

Menu

Statistics > Postestimation

Syntax

After `lasso`, `sqrtlasso`, *and* `elasticnet`

```
lassoselect id = #
```

After `lasso` *and* `sqrtlasso`

```
lassoselect lambda = #
```

After `elasticnet`

```
lassoselect alpha = # lambda = #
```

After `ds` *and* `po` *with* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = # , for(varspec)
```

After `xpo` *without* `resample` *and with* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = # , for(varspec) xfold(#)
```

After `xpo` *with* `resample` *and* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = # , for(varspec) xfold(#) resample(#)
```

After `telasso` *for the outcome variable and with* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = #, for(varspec) tlevel(#)
```

After `telasso` *for the treatment variable and with* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = #, for(varspec)
```

After `telasso` *for the outcome variable with cross-fitting but without* `resample` *and with* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = #, for(varspec) tlevel(#) xfold(#)
```

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

```
lassoselect { id | lambda } = #, for(varspec) xfold(#)
```

After `telasso` *for the outcome variable with cross-fitting and* `resample` *and with* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = #, for(varspec) tlevel(#) xfold(#) resample(#)
```

After `telasso` *for the treatment variable with cross-fitting and* `resample` *and with* `selection(cv)` *or* `selection(adaptive)`

```
lassoselect { id | lambda } = #, for(varspec) xfold(#) resample(#)
```

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

options	Description
* `for(`*varspec*`)`	lasso for *varspec*; `telasso`, `ds`, `po`, and `xpo` commands only
* `xfold(#)`	lasso for the #th cross-fit fold; `xpo` commands and `telasso` with `xfolds` only
* `resample(#)`	lasso for the #th resample; `xpo` commands and `telasso` with `resample` only
* `tlevel(#)`	lasso for the outcome model with the treatment level #; `telasso` only

* `for(`*varspec*`)` is required for all `ds`, `po`, and `xpo` commands and for `telasso`.
`xfold(#)` is required for all `xpo` commands and for `telasso` when the option `xfolds(#)` was specified.
`resample(#)` is required for `xpo` and for `telasso` when the option `resample(#)` was specified.
`tlevel(#)` is required for the outcome model in `telasso`.
`collect` is allowed; see **[U] 11.1.10 Prefix commands**.

Options

`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

stata.com

▷ Example 1: lasso linear

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/r18/fakesurvey_vl
(Fictitious survey data with vl)
. vl rebuild
Rebuilding vl macros ...
```

(*output omitted*)

We want to evaluate our lasso predictions on a sample that we did not use to fit the lasso. So we randomly split our data into two samples of equal sizes. We will fit models on one, and we will use the other to test their predictions. We use `splitsample` to generate a variable indicating the two subsamples.

```
. set seed 1234
. splitsample, generate(sample) nsplit(2)
. label define svalues 1 "Training" 2 "Testing"
. label values sample svalues
```

We fit a lasso linear model on the first subsample.

```
. lasso linear q104 ($idemographics) $ifactors $vlcontinuous
> if sample == 1, rseed(1234)
10-fold cross-validation with 100 lambdas ...
Grid value 1:     lambda = .8978025   no. of nonzero coef. =   4
Folds: 1...5....10   CVF = 16.93341
```

(*output omitted*)

```
Grid value 23:    lambda = .1159557   no. of nonzero coef. =  74
Folds: 1...5....10   CVF = 12.17933
... cross-validation complete ... minimum found
Lasso linear model                          No. of obs        =        458
                                            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	.8978025	4	0.0147	16.93341
18	lambda before	.1846342	42	0.2953	12.10991
* 19	selected lambda	.1682318	49	0.2968	12.08516
20	lambda after	.1532866	55	0.2964	12.09189
23	last lambda	.1159557	74	0.2913	12.17933

* lambda selected by cross-validation.

We store the results because we want to compare these results with other results later.

```
. estimates store lassocv
```

We run lassoknots with options to show the number of nonzero coefficients, estimates of out-of-sample R^2, and the Bayes information criterion (BIC).

```
. lassoknots, display(nonzero osr2 bic)
```

ID	lambda	No. of nonzero coef.	Out-of-sample R-squared	BIC
1	.8978025	4	0.0147	2618.642
2	.8180442	7	0.0236	2630.961
3	.7453714	8	0.0421	2626.254
4	.6791547	9	0.0635	2619.727
5	.6188205	10	0.0857	2611.577
6	.5638462	13	0.1110	2614.155
8	.468115	14	0.1581	2588.189
9	.4265289	16	0.1785	2584.638
10	.3886373	18	0.1980	2580.891
11	.3541118	22	0.2170	2588.984
12	.3226535	26	0.2340	2596.792
13	.2939899	27	0.2517	2586.521
14	.2678726	28	0.2669	2578.211
15	.2440755	32	0.2784	2589.632
16	.2223925	35	0.2865	2593.753
17	.2026358	37	0.2919	2592.923
18	.1846342	42	0.2953	2609.975
* 19	.1682318	49	0.2968	2639.437
20	.1532866	55	0.2964	2663.451
21	.139669	62	0.2952	2693.929
22	.1272612	66	0.2934	2707.174
23	.1159557	74	0.2913	2744.508

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

Research indicates that under certain conditions, selecting the λ that minimizes the BIC gives good predictions. See *BIC* in [LASSO] **lassoknots**.

Here the λ with ID $= 14$ gives the minimum value of the BIC. Let's select it.

```
. lassoselect id = 14
ID = 14  lambda = .2678726 selected
```

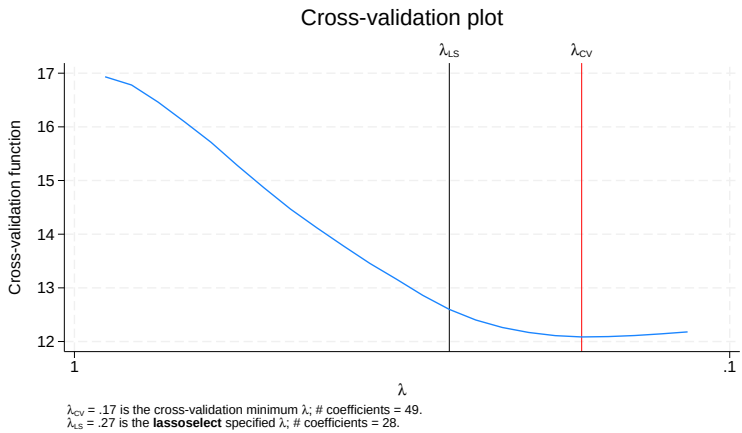
When lassoselect runs, it changes the current estimation results to correspond with the selected lambda. It is almost the same as running another estimation command and wiping out the old estimation results. We say "almost" because it is easy to change λ^* back to what it was originally. We stored our earlier results knowing lassoselect was going to do this.

Let's store the new results from lassoselect.

```
. estimates store lassosel
```

We plot the CV function with the new selected λ^* marked along with the λ selected by cross-validation—the λ that gives the minimum of the CV function.

```
. cvplot
```



The CV function is curving upward at the value of the new selected λ^*. Alternative λ^*'s in a region where the CV function is still relatively flat are sometimes selected, but that is not the case here.

The real test is to see how well it does for out-of-sample prediction compared with the original λ^*. We run `lassogof` to do this.

```
. lassogof lassocv lassosel, over(sample) postselection
Postselection coefficients
```

Name	sample	MSE	R-squared	Obs
lassocv				
	Training	8.652771	0.5065	503
	Testing	14.58354	0.2658	493
lassosel				
	Training	9.740229	0.4421	508
	Testing	13.44496	0.3168	503

The model for λ^* that minimized the BIC did considerably better on out-of-sample prediction than the model for λ^* that minimized the CV function. In-sample prediction was better for the λ^* that minimized the CV function. That is expected because that model contains more variables. But it appears these extra variables were mostly fitting noise, and that hurt the model's out-of-sample predictive ability.

◁

▷ Example 2: dsregress

`lassoselect` can be used after the `ds`, `po`, and `xpo` commands when they are fit using `selection(cv)` or `selection(adaptive)`. See [LASSO] **lasso options**.

We load the data used in [LASSO] **lasso examples**. See that entry for details about the data.

```
. use https://www.stata-press.com/data/r18/fakesurvey_vl, clear
(Fictitious survey data with vl)
. vl rebuild
Rebuilding vl macros ...
  (output omitted)
```

We are going to fit a `dsregress` model with `q104` as our dependent variable and variables of interest `q41` and `q22`. These variables of interest are currently in the variable lists `factors` and `vlcontinuous`, which we will use to specify the control variables. So we need to move them out of these variable lists.

```
. vl modify factors = factors - (q41)
note: 1 variable removed from $factors.
. vl move (q22) vlother
note: 1 variable specified and 1 variable moved.
  (output omitted)
. vl rebuild
Rebuilding vl macros ...
  (output omitted)
```

After we moved the variables out of the variable lists, we typed `vl rebuild` to update the variable list `ifactors` created from `factors`. See [D] **vl** for details.

Before we fit our `dsregress` model using cross-validation, let's fit it using the default `selection(plugin)`.

```
. dsregress q104 i.q41 q22, controls(($idemographics) $ifactors $vlcontinuous)
Estimating lasso for q104 using plugin
Estimating lasso for 1bn.q41 using plugin
Estimating lasso for q22 using plugin
Double-selection linear model         Number of obs               =        914
                                      Number of controls          =        274
                                      Number of selected controls =         33
                                      Wald chi2(2)                =      18.72
                                      Prob > chi2                 =     0.0001
```

q104	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
q41						
Yes	.8410538	.2691082	3.13	0.002	.3136114	1.368496
q22	-.0878443	.0310435	-2.83	0.005	-.1486884	-.0270001

```
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.
```

We run lassoinfo to see how many nonzero coefficients were in each lasso fit by dsregress. It is a good idea to always run lassoinfo after any ds, po, or xpo command.

```
. lassoinfo

    Estimate: active
     Command: dsregress
```

Variable	Model	Selection method	lambda	No. of selected variables
q104	linear	plugin	.1467287	18
1bn.q41	linear	plugin	.1467287	16
q22	linear	plugin	.1467287	15

We now run dsregress with selection(cv),

```
. dsregress q104 i.q41 q22,
> controls(($idemographics) $ifactors $vlcontinuous)
> selection(cv) rseed(1234)

Estimating lasso for q104 using cv
Estimating lasso for 1bn.q41 using cv
Estimating lasso for q22 using cv

Double-selection linear model         Number of obs               =        914
                                      Number of controls          =        274
                                      Number of selected controls =        123
                                      Wald chi2(2)                =      10.96
                                      Prob > chi2                 =     0.0042
```

q104	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
q41						
Yes	.6003918	.2848483	2.11	0.035	.0420994	1.158684
q22	-.0681067	.0306219	-2.22	0.026	-.1281246	-.0080888

```
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.
```

and then run lassoinfo.

```
. lassoinfo

    Estimate: active
     Command: dsregress
```

Variable	Model	Selection method	Selection criterion	lambda	No. of selected variables
q104	linear	cv	CV min.	.1116376	63
1bn.q41	linear	cv	CV min.	.0135958	68
q22	linear	cv	CV min.	.1624043	49

The selection(cv) lassos selected considerably more variables than the selection(plugin) lassos. The CV lassos selected 63, 68, and 49 variables for the lassos, whereas the plugin lassos selected 18, 16, and 15 variables.

We are going to use `lassoselect` to change the selected λ^* for CV lassos to match the number of selected variables in the plugin lassos.

```
. lassoknots, display(nonzero cvmpe osr2) for(q104)
```

ID	lambda	No. of nonzero coef.	CV mean pred. error	Out-of-sample R-squared
1	.864369	4	17.9727	0.0187
2	.7875809	6	17.88282	0.0236
3	.7176144	7	17.64713	0.0365
4	.6538635	8	17.32777	0.0539
5	.595776	12	16.87904	0.0784
6	.5428489	14	16.3203	0.1089
7	.4946237	15	15.74852	0.1401
8	.4506827	18	15.2143	0.1693
(*output omitted*)				
22	.1225221	52	12.02453	0.3435
* 23	.1116376	59	12.02148	0.3436
24	.10172	62	12.02571	0.3434
25	.0926835	71	12.03785	0.3427
26	.0844497	76	12.0626	0.3414
27	.0769474	80	12.09713	0.3395
27	.0769474	80	12.09713	0.3395

```
* lambda selected by cross-validation.
. lassoknots, display(nonzero cvmpe osr2) for(1bn.q41)
```

ID	lambda	No. of nonzero coef.	CV mean pred. error	Out-of-sample R-squared
1	.1155307	4	.2509624	-0.0044
2	.1052673	5	.248763	0.0044
3	.0959156	8	.2442525	0.0224
4	.0873947	9	.2388787	0.0439
5	.0796308	11	.2328436	0.0681
6	.0725566	12	.2262371	0.0945
10	.0500105	15	.2076117	0.1691
12	.0415196	16	.2020617	0.1913
(*output omitted*)				
23	.0149214	61	.1898068	0.2403
* 24	.0135958	64	.1895992	0.2412
25	.012388	68	.1896789	0.2408
26	.0112875	76	.1900733	0.2393
27	.0102847	87	.190537	0.2374
28	.0093711	94	.190995	0.2356

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

```
. lassoknots, display(nonzero cvmpe osr2) for(q22)
```

ID	lambda	No. of nonzero coef.	CV mean pred. error	Out-of-sample R-squared
1	1.380036	4	22.19516	0.0403
2	1.257437	6	21.66035	0.0634
3	1.14573	7	21.01623	0.0913
5	.9512051	8	19.70951	0.1478
9	.6556288	9	18.04511	0.2197
10	.5973845	10	17.74092	0.2329
11	.5443145	11	17.41052	0.2472
12	.4959591	13	17.09005	0.2610
13	.4518995	15	16.78501	0.2742
(*output omitted*)				
23	.1782385	39	14.93049	0.3544
* 24	.1624043	45	14.92344	0.3547
25	.1479767	55	14.93826	0.3541
26	.1348309	67	14.94057	0.3540
27	.1228529	70	14.93962	0.3540
28	.111939	75	14.95101	0.3535

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

When we look at the `lassoinfo` output for the plugin lassos, we see that the value of λ^* for each lasso was the same, namely, 0.1467287. This value does not match up with the same numbers of nonzero coefficients for the CV lassos in these knot tables.

The plugin estimator for λ^* uses estimated coefficient-level weights in its lassos. In theoretical terms, these coefficient-level weights put λ^* on the correct scale for covariate selection by normalizing the scores of the unpenalized estimator. In practical terms, these weights cause the effective scale of λ for `selection(plugin)` and `selection(cv)` to differ.

We select the λ^*'s for each CV lasso to match the number of nonzero coefficients of the plugin lassos.

```
. lassoselect id = 6, for(q104)
ID = 6  lambda = .5428489 selected
. lassoselect id = 6, for(1bn.q41)
ID = 6  lambda = .0725566 selected
. lassoselect id = 11, for(q22)
ID = 11  lambda = .5443145 selected
```

To update our dsregress model with these new λ^*'s, we rerun the command with the reestimate option. Then, we run lassoinfo to confirm that the lassos produced the same number of nonzero coefficients.

```
. dsregress, reestimate

Double-selection linear model          Number of obs               =        914
                                       Number of controls          =        274
                                       Number of selected controls =         33
                                       Wald chi2(2)                =      18.72
                                       Prob > chi2                 =     0.0001
```

q104	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
q41						
Yes	.8410538	.2691082	3.13	0.002	.3136114	1.368496
q22	-.0878443	.0310435	-2.83	0.005	-.1486884	-.0270001

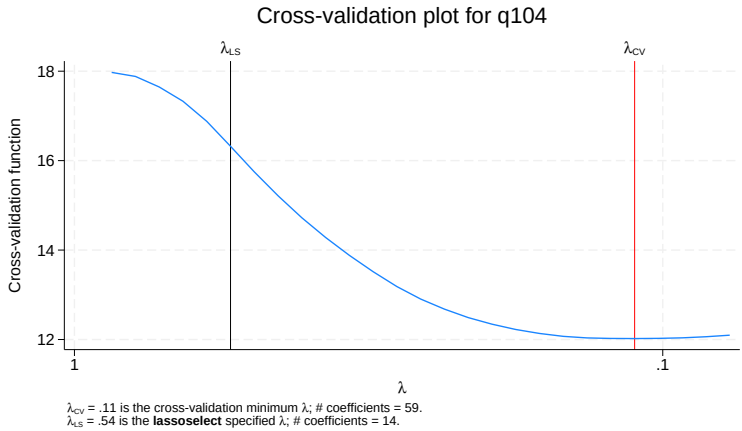
```
Note: Chi-squared test is a Wald test of the coefficients of the variables
      of interest jointly equal to zero. Lassos select controls for model
      estimation. Type lassoinfo to see number of selected variables in each
      lasso.

. lassoinfo

    Estimate: active
     Command: dsregress
```

Variable	Model	Selection method	Selection criterion	lambda	No. of selected variables
q104	linear	user	user	.5428489	18
1bn.q41	linear	user	user	.0725566	16
q22	linear	user	user	.5443145	15

These new dsregress results are exactly the same as the dsregress results produced with plugin lassos.

We can plot the CV function and see where the new λ^* falls. We do so for the lasso for the dependent variable q104.

```
. cvplot, for(q104)
```



It may be that the plugin lassos underselected controls for this problem. Or it may be that the plugin lassos actually did fine and the CV lassos overselected controls. We might want to continue these sensitivity analyses and pick some λ^*'s intermediate between the plugin values and the CV values. Plugin selection and CV selection are not just two different numerical techniques, they are two different modeling techniques, each with a different set of assumptions. See [LASSO] **Inference requirements**.

Stored results

lassoselect stores the following in r():

Macros

r(varlist)	selected variables

Also see

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

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