lassoselect — Select lambda after lasso

Description

lassoselect allows the user to select a different $\lambda^*$ after \texttt{lasso} and \texttt{sqrtlasso} when the selection method was selection(cv), selection(adaptive), or selection(none).

After \texttt{elasticnet}, the user can select a different $(\alpha^*, \lambda^*)$ pair.

When the \texttt{ds}, \texttt{po}, and \texttt{xpo} commands fit models using selection(cv) or selection(adaptive) ([\texttt{LASSO}] \texttt{lasso options}), lassoselect can be used to select a different $\lambda^*$ for a particular lasso.

Quick start

After \texttt{lasso} with selection(cv), change the selected $\lambda^*$ to that with ID = 52
\begin{verbatim}
lassoselect id = 52
\end{verbatim}

As above, but change the selected $\lambda^*$ to the $\lambda$ closest to 0.01
\begin{verbatim}
lassoselect lambda = 0.01
\end{verbatim}

After \texttt{elasticnet}, change the selected $(\alpha^*, \lambda^*)$ to $(0.5, 0.267345)$
\begin{verbatim}
lassoselect alpha = 0.5 lambda = 0.267345
\end{verbatim}

After \texttt{dsregress} with selection(adaptive), change the selected $\lambda^*$ to 1.65278 for the adaptive lasso for the variable $y$
\begin{verbatim}
lassoselect lambda = 1.65278, for(y)
\end{verbatim}

After \texttt{poivregress} with selection(cv), change the selected $\lambda^*$ to the $\lambda$ closest to 0.7 for the lasso for the prediction of the variable income
\begin{verbatim}
lassoselect lambda = 0.7, for(pred(income))
\end{verbatim}

After \texttt{xporegress} with selection(cv) and resample, change the selected $\lambda^*$ to 0.234189 for the lasso for the variable $x_{26}$ for the 5th cross-fit fold in the 9th resample
\begin{verbatim}
lassoselect lambda = 0.234189, for(x26) xfold(5) resample(9)
\end{verbatim}

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(#)

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

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*for(varspec)</td>
<td>lasso for varspec; ds, po, and xpo commands only</td>
</tr>
<tr>
<td>*xfold(#)</td>
<td>lasso for the #th cross-fit fold; xpo commands only</td>
</tr>
<tr>
<td>*resample(#)</td>
<td>lasso for the #th resample; xpo commands with resample only</td>
</tr>
</tbody>
</table>

*for(varspec) is required for all ds, po, and xpo commands. xfold(#) is required for all xpo commands. resample(#) is required for xpo when the option resample(#) was specified.

Options

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

This option is required after ds, po, and xpo commands.
xfold(#) specifies a particular lasso after an xpo estimation command. 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 \( # = 1, 2, \ldots, K \). It is required after an xpo command.

resample(#) specifies a particular lasso after an xpo estimation command 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 \( # = 1, 2, \ldots, R \). This option, along with xfold(#), is required after an xpo command with resampling.

Remarks and examples

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/r16/fakesurvey_vl
   (Fictitious Survey Data with vl)
. vl rebuild
```

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 = .9109571 no. of nonzero coef. = 4
  Folds: 1...5....10 CVF = 16.93341

(output omitted)

Grid value 23: lambda = .1176546 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 = 273

Selection: Cross-validation
  No. of CV folds = 10

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>lambda</th>
<th>nonzeros</th>
<th>R-squared</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>first lambda</td>
<td>.9109571</td>
<td>4</td>
<td>0.0147</td>
<td>16.93341</td>
</tr>
<tr>
<td>18</td>
<td>lambda before</td>
<td>.1873395</td>
<td>42</td>
<td>0.2953</td>
<td>12.10991</td>
</tr>
<tr>
<td>* 19</td>
<td>selected lambda</td>
<td>.1706967</td>
<td>49</td>
<td>0.2968</td>
<td>12.08516</td>
</tr>
<tr>
<td>20</td>
<td>lambda after</td>
<td>.1555325</td>
<td>55</td>
<td>0.2964</td>
<td>12.09189</td>
</tr>
<tr>
<td>23</td>
<td>last lambda</td>
<td>.1176546</td>
<td>74</td>
<td>0.2913</td>
<td>12.17933</td>
</tr>
</tbody>
</table>

* 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)
```

<table>
<thead>
<tr>
<th>ID</th>
<th>lambda</th>
<th>No. of nonzero coef.</th>
<th>Out-of-sample R-squared</th>
<th>BIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.9109571</td>
<td>4</td>
<td>0.0147</td>
<td>2618.642</td>
</tr>
<tr>
<td>2</td>
<td>.8300302</td>
<td>7</td>
<td>0.0236</td>
<td>2630.961</td>
</tr>
<tr>
<td>3</td>
<td>.7562926</td>
<td>8</td>
<td>0.0421</td>
<td>2626.254</td>
</tr>
<tr>
<td>4</td>
<td>.6891057</td>
<td>9</td>
<td>0.0635</td>
<td>2619.727</td>
</tr>
<tr>
<td>5</td>
<td>.6278874</td>
<td>10</td>
<td>0.0857</td>
<td>2611.577</td>
</tr>
<tr>
<td>6</td>
<td>.5721076</td>
<td>13</td>
<td>0.1110</td>
<td>2614.155</td>
</tr>
<tr>
<td>7</td>
<td>.4749738</td>
<td>14</td>
<td>0.1581</td>
<td>2588.189</td>
</tr>
<tr>
<td>8</td>
<td>.4327784</td>
<td>16</td>
<td>0.1785</td>
<td>2584.638</td>
</tr>
<tr>
<td>9</td>
<td>.3943316</td>
<td>18</td>
<td>0.1980</td>
<td>2580.891</td>
</tr>
<tr>
<td>10</td>
<td>.3593003</td>
<td>22</td>
<td>0.2170</td>
<td>2588.984</td>
</tr>
<tr>
<td>11</td>
<td>.327381</td>
<td>26</td>
<td>0.2340</td>
<td>2596.792</td>
</tr>
<tr>
<td>12</td>
<td>.2982974</td>
<td>27</td>
<td>0.2517</td>
<td>2586.521</td>
</tr>
<tr>
<td>13</td>
<td>.2717975</td>
<td>28</td>
<td>0.2669</td>
<td>2578.211</td>
</tr>
<tr>
<td>14</td>
<td>.2476517</td>
<td>32</td>
<td>0.2784</td>
<td>2589.632</td>
</tr>
<tr>
<td>15</td>
<td>.225651</td>
<td>35</td>
<td>0.2865</td>
<td>2593.753</td>
</tr>
<tr>
<td>16</td>
<td>.2056048</td>
<td>37</td>
<td>0.2919</td>
<td>2592.923</td>
</tr>
<tr>
<td>17</td>
<td>.1873395</td>
<td>42</td>
<td>0.2953</td>
<td>2609.975</td>
</tr>
<tr>
<td>18</td>
<td>.1706967</td>
<td>49</td>
<td>0.2968</td>
<td>2639.437</td>
</tr>
<tr>
<td>19</td>
<td>.1555325</td>
<td>55</td>
<td>0.2964</td>
<td>2663.451</td>
</tr>
<tr>
<td>20</td>
<td>.1417154</td>
<td>62</td>
<td>0.2952</td>
<td>2693.929</td>
</tr>
<tr>
<td>21</td>
<td>.1291258</td>
<td>66</td>
<td>0.2934</td>
<td>2707.174</td>
</tr>
<tr>
<td>22</td>
<td>.1176546</td>
<td>74</td>
<td>0.2913</td>
<td>2744.508</td>
</tr>
</tbody>
</table>

* lambda selected by cross-validation.

Research indicates that under certain conditions, selecting the $\lambda$ that minimizes the BIC gives good predictions. See `BIC` in [LASSO] `lassoknots`.

Here the $\lambda$ with ID $= 14$ gives the minimum value of the BIC. Let’s select it.

```
. lassoselect id = 14
ID = 14  lambda = .2717975 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 $\lambda^*$ 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 $\lambda^*$ marked along with the $\lambda$ selected by cross-validation—the $\lambda$ that gives the minimum of the CV function.

```
. cvplot
```

![Cross-validation plot](image)

The CV function is curving upward at the value of the new selected $\lambda^*$. Alternative $\lambda^*$'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 $\lambda^*$. We run `lassogof` to do this.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>sample</th>
<th>MSE</th>
<th>R-squared</th>
<th>Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>lassocv</td>
<td>Training</td>
<td>8.652771</td>
<td>0.5065</td>
<td>503</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>14.58354</td>
<td>0.2658</td>
<td>493</td>
</tr>
<tr>
<td>lassosel</td>
<td>Training</td>
<td>9.740229</td>
<td>0.4421</td>
<td>508</td>
</tr>
<tr>
<td></td>
<td>Testing</td>
<td>13.44496</td>
<td>0.3168</td>
<td>503</td>
</tr>
</tbody>
</table>

The model for $\lambda^*$ that minimized the BIC did considerably better on out-of-sample prediction than the model for $\lambda^*$ that minimized the CV function. In-sample prediction was better for the $\lambda^*$ 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/r16/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 = 29
   Wald chi2(2) = 18.72
   Prob > chi2 = 0.0001

|           | Coefficient | Std. Err. | z     | P>|z|  | [95% Conf. Interval] |
|-----------|-------------|-----------|-------|------|----------------------|
| q104      |             |           |       |      |                      |
| 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Selection method</th>
<th>lambda</th>
<th>No. of selected variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>q104</td>
<td>linear plugin</td>
<td>.1467287</td>
<td>14</td>
</tr>
<tr>
<td>1bn.q41</td>
<td>linear plugin</td>
<td>.1467287</td>
<td>12</td>
</tr>
<tr>
<td>q22</td>
<td>linear plugin</td>
<td>.1467287</td>
<td>11</td>
</tr>
</tbody>
</table>
```

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 = 119
Wald chi2(2) = 10.96
Prob > chi2 = 0.0042

| Variable | Coef. | Std. Err. | z     | P>|z|   | [95% Conf. Interval] |
|----------|-------|-----------|-------|-------|---------------------|
| q41      | .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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Selection method</th>
<th>Selection criterion</th>
<th>lambda</th>
<th>No. of selected variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>q104</td>
<td>linear cv</td>
<td>CV min.</td>
<td>.1132914</td>
<td>59</td>
</tr>
<tr>
<td>1bn.q41</td>
<td>linear cv</td>
<td>CV min.</td>
<td>.0137972</td>
<td>64</td>
</tr>
<tr>
<td>q22</td>
<td>linear cv</td>
<td>CV min.</td>
<td>.1648102</td>
<td>45</td>
</tr>
</tbody>
</table>
```

The `selection(cv)` lassos selected considerably more variables than the `selection(plugin)` lassos. The CV lassos selected 62, 61, and 45 variables for the lassos, whereas the plugin lassos selected 14, 12, and 11 variables.
We are going to use `lassoselect` to change the selected $\lambda^*$ for CV lassos to match the number of selected variables in the plugin lassos.

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

<table>
<thead>
<tr>
<th>ID</th>
<th>lambda</th>
<th>No. of nonzero coef.</th>
<th>CV mean pred.</th>
<th>Out-of-sample error</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8771745</td>
<td>4</td>
<td>17.9727</td>
<td>0.0187</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.7992487</td>
<td>6</td>
<td>17.8822</td>
<td>0.0236</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.7282457</td>
<td>7</td>
<td>17.6471</td>
<td>0.0365</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.6635503</td>
<td>8</td>
<td>17.3277</td>
<td>0.0539</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.6046024</td>
<td>12</td>
<td>16.8790</td>
<td>0.0784</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.5508912</td>
<td>14</td>
<td>16.3203</td>
<td>0.1089</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.5019515</td>
<td>15</td>
<td>15.7485</td>
<td>0.1401</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.4573595</td>
<td>18</td>
<td>15.2143</td>
<td>0.1693</td>
<td></td>
</tr>
</tbody>
</table>

(output omitted)

```
. lassoknots, display(nonzero cvmpe osr2) for(lbn.q41)
```

<table>
<thead>
<tr>
<th>ID</th>
<th>lambda</th>
<th>No. of nonzero coef.</th>
<th>CV mean pred.</th>
<th>Out-of-sample error</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1172423</td>
<td>4</td>
<td>2509624</td>
<td>0.0044</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.1068268</td>
<td>5</td>
<td>248763</td>
<td>0.0044</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.0973366</td>
<td>8</td>
<td>2442525</td>
<td>0.0224</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0886895</td>
<td>9</td>
<td>2388787</td>
<td>0.0439</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.0808105</td>
<td>11</td>
<td>2328436</td>
<td>0.0681</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0736316</td>
<td>12</td>
<td>2262371</td>
<td>0.0945</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.0507514</td>
<td>15</td>
<td>2076117</td>
<td>0.1691</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.0421347</td>
<td>16</td>
<td>2020617</td>
<td>0.1913</td>
<td></td>
</tr>
</tbody>
</table>

(output omitted)

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

```
. lassoknots, display(nonzero cvmpe osr2) for(lbn.q41)
```

<table>
<thead>
<tr>
<th>ID</th>
<th>lambda</th>
<th>No. of nonzero coef.</th>
<th>CV mean pred.</th>
<th>Out-of-sample error</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>0.0151424</td>
<td>61</td>
<td>1898068</td>
<td>0.2403</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.0137972</td>
<td>64</td>
<td>1895992</td>
<td>0.2412</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.0125715</td>
<td>68</td>
<td>1896789</td>
<td>0.2408</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.0114547</td>
<td>76</td>
<td>1900733</td>
<td>0.2393</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>0.0104371</td>
<td>87</td>
<td>190537</td>
<td>0.2374</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>0.0095099</td>
<td>94</td>
<td>190995</td>
<td>0.2356</td>
<td></td>
</tr>
</tbody>
</table>

* lambda selected by cross-validation.
When we look at the lassoinfo output for the plugin lassos, we see that the value of $\lambda^*$ 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 $\lambda^*$ uses estimated coefficient-level weights in its lassos. In theoretical terms, these coefficient-level weights put $\lambda^*$ 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 $\lambda$ for selection(plugin) and selection(cv) to differ.

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

```
. lassoselect id = 6, for(q104)
ID = 6 lambda = .5508912 selected
. lassoselect id = 6, for(1bn.q41)
ID = 6 lambda = .0736316 selected
. lassoselect id = 11, for(q22)
ID = 11 lambda = .5523785 selected
```
To update our dsregress model with these new $\lambda^*$'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 = 29
Wald chi2(2) = 18.72
Prob > chi2 = 0.0001

                     Robust
                     Coef. Std. Err. z P>|z| [95% Conf. Interval]
q104
  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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Selection Model</th>
<th>Selection method</th>
<th>Selection criterion</th>
<th>lambda</th>
<th>No. of selected variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>q104</td>
<td>linear</td>
<td>user</td>
<td>user</td>
<td>.5508912</td>
<td>14</td>
</tr>
<tr>
<td>1bn.q41</td>
<td>linear</td>
<td>user</td>
<td>user</td>
<td>.0736316</td>
<td>12</td>
</tr>
<tr>
<td>q22</td>
<td>linear</td>
<td>user</td>
<td>user</td>
<td>.5523785</td>
<td>11</td>
</tr>
</tbody>
</table>
```

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 $\lambda^*$ falls. We do so for the lasso for the dependent variable $q_{104}$.

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

![Cross-validation plot for q104](image)

$\lambda_{CV}$ Cross-validation minimum lambda. $\lambda_{LS} = .11$, # Coefficients=59. $\lambda_{LS}$ lassoselect specified lambda. $\lambda = .55$, # Coefficients=14.

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 $\lambda^*$’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