Stata Webinar

Introduction to lasso using Stata

Miguel Dorta

StataCorp LLC

June 28, 2023

Outline

- Overview of lasso in Stata

Outline

- Overview of lasso in Stata
- Lasso for prediction and model selection
    - Motivation and basic theoretical aspects
    - Example for a linear model
        - Basic workflow
        - Some tools and options

Outline

- Overview of lasso in Stata
- Lasso for prediction and model selection
  - Motivation and basic theoretical aspects
  - Example for a linear model
    - Basic workflow
    - Some tools and options
- Lasso for inference
  - Motivation and basic theoretical aspects
  - Example for a linear model
    - Basic estimations
    - Some tools and options

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **`lasso`** for performing lasso (linear, logit, probit, Poisson, and Cox)

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **`lasso`** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **`elasticnet`** for performing elastic net (linear, logit, probit, Poisson, and Cox)

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **`lasso`** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **`elasticnet`** for performing elastic net (linear, logit, probit, Poisson, and Cox)
  - **`sqrtlasso`** for performing square-root lasso (linear)

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **lasso** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **elasticnet** for performing elastic net (linear, logit, probit, Poisson, and Cox)
  - **sqrtlasso** for performing square-root lasso (linear)
- Lasso estimators for inference
  - Double-selection method: **dsregress**, **dslogit**, and **dspoisson**

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **lasso** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **elasticnet** for performing elastic net (linear, logit, probit, Poisson, and Cox)
  - **sqrtlasso** for performing square-root lasso (linear)
- Lasso estimators for inference
  - Double-selection method: **dsregress**, **dslogit**, and **dspoisson**
  - Partialing-out method: **poregress**, **poivregress**, **pologit**, and **popoisson**

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **lasso** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **elasticnet** for performing elastic net (linear, logit, probit, Poisson, and Cox)
  - **sqrtlasso** for performing square-root lasso (linear)
- Lasso estimators for inference
  - Double-selection method: **dsregress**, **dslogit**, and **dspoisson**
  - Partialing-out method: **poregress**, **poivregress**, **pologit**, and **popoisson**
  - Cross-fit partialing-out method (double machine learning): **xporegress**, **xpoivregress**, **xpologit**, and **xpopoisson**

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **lasso** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **elasticnet** for performing elastic net (linear, logit, probit, Poisson, and Cox)
  - **sqrtlasso** for performing square-root lasso (linear)
- Lasso estimators for inference
  - Double-selection method: **dsregress**, **dslogit**, and **dspoisson**
  - Partialing-out method: **poregress**, **poivregress**, **pologit**, and **popoisson**
  - Cross-fit partialing-out method (double machine learning): **xporegress**, **xpoivregress**, **xpologit**, and **xpopoisson**
  - Treatment effects estimation using lasso: **telasso**

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **`lasso`** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **`elasticnet`** for performing elastic net (linear, logit, probit, Poisson, and Cox)
  - **`sqrtlasso`** for performing square-root lasso (linear)
- Lasso estimators for inference
  - Double-selection method: **`dsregress`**, **`dslogit`**, and **`dspoisson`**
  - Partialing-out method: **`poregress`**, **`poivregress`**, **`pologit`**, and **`popoisson`**
  - Cross-fit partialing-out method (double machine learning): **`xporegress`**, **`xpoivregress`**, **`xpologit`**, and **`xpopoisson`**
  - Treatment effects estimation using lasso: **`telasso`**
- Most lasso features are available from Stata 16. **`telasso`**, selection using BIC, and clustering were added in Stata 17. Cox models with **`lasso`** and **`elasticnet`** were added in Stata 18

Overview of lasso in Stata

- Lasso for prediction and model selection
  - **lasso** for performing lasso (linear, logit, probit, Poisson, and Cox)
  - **elasticnet** for performing elastic net (linear, logit, probit, Poisson, and Cox)
  - **sqrtlasso** for performing square-root lasso (linear)
- Lasso estimators for inference
  - Double-selection method: **dsregress**, **dslogit**, and **dspoisson**
  - Partialing-out method: **poregress**, **poivregress**, **pologit**, and **popoisson**
  - Cross-fit partialing-out method (double machine learning): **xporegress**, **xpoivregress**, **xpologit**, and **xpopoisson**
  - Treatment effects estimation using lasso: **telasso**
- Most lasso features are available from Stata 16. **telasso**, selection using BIC, and clustering were added in Stata 17. Cox models with **lasso** and **elasticnet** were added in Stata 18
- "Lasso was an acronym for 'least absolute shrinkage and selection operator'. Today, lasso is considered a word"

Lasso for prediction and model selection

Motivation

- Suppose we have original data with an outcome variable and lots of covariates

Motivation

- Suppose we have original data with an outcome variable and lots of covariates
- We then have new data on the covariates but not on the outcome variable

Motivation

- Suppose we have original data with an outcome variable and lots of covariates
- We then have new data on the covariates but not on the outcome variable
- We want to predict the outcome variable on the new data

Motivation

- Suppose we have original data with an outcome variable and lots of covariates
- We then have new data on the covariates but not on the outcome variable
- We want to predict the outcome variable on the new data
- So, we fit a model for the outcome variable on some of the covariates using the original data

Motivation

- Suppose we have original data with an outcome variable and lots of covariates
- We then have new data on the covariates but not on the outcome variable
- We want to predict the outcome variable on the new data
- So, we fit a model for the outcome variable on some of the covariates using the original data
- The best prediction minimizes the mean-squared error or another loss function on new data

Motivation

- Suppose we have original data with an outcome variable and lots of covariates
- We then have new data on the covariates but not on the outcome variable
- We want to predict the outcome variable on the new data
- So, we fit a model for the outcome variable on some of the covariates using the original data
- The best prediction minimizes the mean-squared error or another loss function on new data
- For example, a car dealer business needs to predict the market price of used cars, given many potential predictor variables

Motivation

- Suppose we have original data with an outcome variable and lots of covariates
- We then have new data on the covariates but not on the outcome variable
- We want to predict the outcome variable on the new data
- So, we fit a model for the outcome variable on some of the covariates using the original data
- The best prediction minimizes the mean-squared error or another loss function on new data
- For example, a car dealer business needs to predict the market price of used cars, given many potential predictor variables

- If data have lots of covariates, which ones should we include in our prediction model?

Using penalized regression to avoid overfitting

- Problems if all potential covariates would be included:
  - It would not be possible if $p > N$
  - Even $p < N$, too many covariates may produce overfitting

Using penalized regression to avoid overfitting

- Problems if all potential covariates would be included:
  - It would not be possible if $p > N$
  - Even $p < N$, too many covariates may produce overfitting
  - Overfitting is the presence of excessive parameters that improve the in-sample prediction performance but worsen the out-of-sample performance

Using penalized regression to avoid overfitting

- Problems if all potential covariates would be included:
  - It would not be possible if p > N
  - Even p < N, too many covariates may produce overfitting
  - Overfitting is the presence of excessive parameters that improve the in-sample prediction performance but worsen the out-of-sample performance
- A way to avoid overfitting is by penalizing the objective function

$$Q = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\beta')$$

- $f(.)$ is a measure of prediction error

Using penalized regression to avoid overfitting

- Problems if all potential covariates would be included:
  - It would not be possible if p > N
  - Even p < N, too many covariates may produce overfitting
  - Overfitting is the presence of excessive parameters that improve the in-sample prediction performance but worsen the out-of-sample performance
- A way to avoid overfitting is by penalizing the objective function

$$Q = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i \beta')$$

- $f(.)$ is a measure of prediction error
- How does lasso penalize the objective function?

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\beta') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- λ is the lasso penalty parameter

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- λ is the lasso penalty parameter
- Covariates in $\boldsymbol{x}$ are standardized

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- λ is the lasso penalty parameter
- Covariates in $\boldsymbol{x}$ are standardized
- Given a value of λ, minimizing Q_L causes shrinkage in the β_j's towards zero

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- λ is the lasso penalty parameter
- Covariates in $\boldsymbol{x}$ are standardized
- Given a value of λ, minimizing Q_L causes shrinkage in the β_j's towards zero
- The kink in the absolute value shrinks some of the $\beta_j's$ to zero
  - covariates with $\beta_j = 0$ are excluded
  - covariates with $\beta_j \neq 0$ are included

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- λ is the lasso penalty parameter
- Covariates in $\boldsymbol{x}$ are standardized
- Given a value of λ, minimizing Q_L causes shrinkage in the β_j's towards zero
- The kink in the absolute value shrinks some of the $\beta_j's$ to zero
  - covariates with $\beta_j = 0$ are excluded
  - covariates with $\beta_j \neq 0$ are included
- The minimization algorithm is called "coordinate descent"

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i \boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j |\beta_j|$$

- λ is the lasso penalty parameter
- Covariates in $\boldsymbol{x}$ are standardized
- Given a value of λ, minimizing Q_L causes shrinkage in the β_j's towards zero
- The kink in the absolute value shrinks some of the $\beta_j's$ to zero
  - covariates with $\beta_j = 0$ are excluded
  - covariates with $\beta_j \neq 0$ are included
- The minimization algorithm is called "coordinate descent"
- Given a dataset, there exists a λ_{max} that shrinks all the coefficients to zero

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\beta') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- λ is the lasso penalty parameter
- Covariates in $\boldsymbol{x}$ are standardized
- Given a value of λ, minimizing Q_L causes shrinkage in the β_j's towards zero
- The kink in the absolute value shrinks some of the $\beta_j's$ to zero
  - covariates with $\beta_j = 0$ are excluded
  - covariates with $\beta_j \neq 0$ are included
- The minimization algorithm is called "coordinate descent"
- Given a dataset, there exists a λ_{max} that shrinks all the coefficients to zero
- As λ decreases, more variables are selected

Using penalized regression to avoid over-fitting

Lasso (Tibshirani, 1996) minimizes the penalized objective function

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- λ is the lasso penalty parameter
- Covariates in ***x*** are standardized
- Given a value of λ, minimizing Q_L causes shrinkage in the β_j's towards zero
- The kink in the absolute value shrinks some of the $\beta_j's$ to zero
  - covariates with $\beta_j = 0$ are excluded
  - covariates with $\beta_j \neq 0$ are included
- The minimization algorithm is called "coordinate descent"
- Given a dataset, there exists a λ_{max} that shrinks all the coefficients to zero
- As λ decreases, more variables are selected
- Least absolute shrinkage and selection operator (lasso)

Using penalized regression to avoid overfitting

- Penalized objective function for lasso

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\beta') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

Using penalized regression to avoid overfitting

- Penalized objective function for lasso

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- Penalized objective function for elastic net

$$Q_{en} = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j \left\{\frac{1-\alpha}{2}\beta_j^2 + \alpha|\beta_j|\right\}$$

Using penalized regression to avoid overfitting

- Penalized objective function for lasso

$$Q_L = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j|\beta_j|$$

- Penalized objective function for elastic net

$$Q_{en} = \frac{1}{N}\sum_{i=1}^{N} w_i f(y_i, \beta_0 + \mathbf{x}_i\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} k_j \left\{\frac{1-\alpha}{2}\beta_j^2 + \alpha|\beta_j|\right\}$$

- Penalized objective function for square-root lasso

$$Q_L = \sqrt{\frac{1}{N}\sum_{i=1}^{N} w_i(y_i - \beta_0 - \mathbf{x}_i\boldsymbol{\beta}')^2} + \frac{\lambda}{N}\sum_{j=1}^{p} k_j|\beta_j|$$

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda's$ (default is 100 $\lambda's$) such that the largest λ does not select any variables (all of the coefficients are zero)

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda's$ (default is 100 $\lambda's$) such that the largest λ does not select any variables (all of the coefficients are zero)
2. divide the data randomly into K partitions called folds (default is 10 folds)

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda's$ (default is 100 $\lambda's$) such that the largest λ does not select any variables (all of the coefficients are zero)
2. divide the data randomly into K partitions called folds (default is 10 folds)
3. given a value of λ, fit the lasso on all observations except those in fold k

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda's$ (default is 100 $\lambda's$) such that the largest λ does not select any variables (all of the coefficients are zero)
2. divide the data randomly into K partitions called folds (default is 10 folds)
3. given a value of λ, fit the lasso on all observations except those in fold k
4. that result is used to predict the outcome in fold k and a measure of prediction error is computed

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda' s$ (default is 100 $\lambda' s$) such that the largest λ does not select any variables (all of the coefficients are zero)
2. divide the data randomly into K partitions called folds (default is 10 folds)
3. given a value of λ, fit the lasso on all observations except those in fold k
4. that result is used to predict the outcome in fold k and a measure of prediction error is computed
5. steps 3 and 4 are repeated for each fold

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda's$ (default is 100 $\lambda's$) such that the largest λ does not select any variables (all of the coefficients are zero)
2. divide the data randomly into K partitions called folds (default is 10 folds)
3. given a value of λ, fit the lasso on all observations except those in fold k
4. that result is used to predict the outcome in fold k and a measure of prediction error is computed
5. steps 3 and 4 are repeated for each fold
6. the prediction errors are then averaged over all folds

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda's$ (default is 100 $\lambda's$) such that the largest λ does not select any variables (all of the coefficients are zero)
2. divide the data randomly into K partitions called folds (default is 10 folds)
3. given a value of λ, fit the lasso on all observations except those in fold k
4. that result is used to predict the outcome in fold k and a measure of prediction error is computed
5. steps 3 and 4 are repeated for each fold
6. the prediction errors are then averaged over all folds
7. steps 3, 4, 5 and 6 are repeated for each λ in the grid

Selecting λ using K-fold cross-validation

This is the default selection method for λ in lasso for prediction

1. compute a grid of $\lambda' s$ (default is 100 $\lambda' s$) such that the largest λ does not select any variables (all of the coefficients are zero)
2. divide the data randomly into K partitions called folds (default is 10 folds)
3. given a value of λ, fit the lasso on all observations except those in fold k
4. that result is used to predict the outcome in fold k and a measure of prediction error is computed
5. steps 3 and 4 are repeated for each fold
6. the prediction errors are then averaged over all folds
7. steps 3, 4, 5 and 6 are repeated for each λ in the grid
8. select the $\lambda*$ with the smallest average prediction error, and refit lasso using $\lambda*$ on the original data

Example on lasso for prediction with a linear model

- Predicting infant birth weight

Example on lasso for prediction with a linear model

- Predicting infant birth weight
- Basic covariates: 5 binary variables and 6 continuous variables of mothers and fathers
- Covariates: main effects and interactions (117 covariates)

Example on lasso for prediction with a linear model

- Predicting infant birth weight
- Basic covariates: 5 binary variables and 6 continuous variables of mothers and fathers
- Covariates: main effects and interactions (117 covariates)
- Number of observations: 4642

Example on lasso for prediction with a linear model

- Predicting infant birth weight
- Basic covariates: 5 binary variables and 6 continuous variables of mothers and fathers
- Covariates: main effects and interactions (117 covariates)
- Number of observations: 4642
- Among OLS, lasso, elastic-net, and square-root lasso, which method should be used to predict the infant birth weight?

Workflow

- Step 1: Using `splitsample`
  - Evaluate lasso predictions on a sub-sample that we did not use to fit the lasso. So, we randomly split data into training and testing sample

Workflow

- Step 1: Using **`splitsample`**
  - Evaluate lasso predictions on a sub-sample that we did not use to fit the lasso. So, we randomly split data into training and testing sample

```
 . use cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138154)
. set seed 1907
. splitsample, generate(sample) split(0.70 0.30)
. label define lbsample 1 "Training" 2 "Testing"
. label value sample lbsample
```

Workflow

- Step 1: Using **`splitsample`**
  - Evaluate lasso predictions on a sub-sample that we did not use to fit the lasso. So, we randomly split data into training and testing sample

```
 . use cattaneo2
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138154)
. set seed 1907
. splitsample, generate(sample) split(0.70 0.30)
. label define lbsample 1 "Training" 2 "Testing"
. label value sample lbsample
```

- Step 2: Create macro with factor variable syntax

```
 . global covs (c.mage c.fage c.mage c.fage c.monthslb)##(c.mage ///
>       c.fage c.mage c.fage c.monthslb) (mmarried mhisp fhisp ///
>           foreign alcohol msmoke fbaby prenatal1)##(c.mage c.fage ///
>           c.medu c.fedu c.monthslb)
```

Workflow

- Step 3: Select λ parameter value using training sample

```
. quietly regress bweight $covs if sample == 1
. estimates store ols
. quietly lasso linear bweight $covs if sample == 1
. estimates store lasso
. quietly elasticnet linear bweight $covs if sample == 1, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnet
. quietly sqrtlasso bweight $covs if sample == 1
. estimates store sqlasso
```

Workflow

- Step 3: Select λ parameter value using training sample

```
. quietly regress bweight $covs if sample == 1
. estimates store ols
. quietly lasso linear bweight $covs if sample == 1
. estimates store lasso
. quietly elasticnet linear bweight $covs if sample == 1, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnet
. quietly sqrtlasso bweight $covs if sample == 1
. estimates store sqlasso
```

- **`if sample == 1`** restricts commands to use only the training data

Workflow

- Step 3: Select λ parameter value using training sample

```
. quietly regress bweight $covs if sample == 1
. estimates store ols
. quietly lasso linear bweight $covs if sample == 1
. estimates store lasso
. quietly elasticnet linear bweight $covs if sample == 1, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnet
. quietly sqrtlasso bweight $covs if sample == 1
. estimates store sqlasso
```

  - **`if sample == 1`** restricts commands to use only the training data
  - By default, λ is chosen by 10-fold cross-validation (grid of 100 values of λ)

Workflow

- Step 3: Select λ parameter value using training sample

```
. quietly regress bweight $covs if sample == 1
. estimates store ols
. quietly lasso linear bweight $covs if sample == 1
. estimates store lasso
. quietly elasticnet linear bweight $covs if sample == 1, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnet
. quietly sqrtlasso bweight $covs if sample == 1
. estimates store sqlasso
```

  - **`if sample == 1`** restricts commands to use only the training data
  - By default, λ is chosen by 10-fold cross-validation (grid of 100 values of λ)
  - **`estimates store`** stores estimation results

Workflow

- Step 3: Select λ parameter value using training sample

```
. quietly regress bweight $covs if sample == 1
. estimates store ols
. quietly lasso linear bweight $covs if sample == 1
. estimates store lasso
. quietly elasticnet linear bweight $covs if sample == 1, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnet
. quietly sqrtlasso bweight $covs if sample == 1
. estimates store sqlasso
```

- **`if sample == 1`** restricts commands to use only the training data
- By default, λ is chosen by 10-fold cross-validation (grid of 100 values of λ)
- **`estimates store`** stores estimation results
- In **`elasticnet`**, option **`alpha()`** specifies some α values for the penalty term

Workflow

- Step 4: Evaluate prediction performance using testing sample

```
. lassogof ols lasso elastnet sqlasso, over(sample)
Penalized coefficients
```

Name	sample	MSE	R-squared	Obs
ols				
	Training	304368.1	0.0800	3,249
	Testing	328554.5	0.0463	1,393
lasso				
	Training	310573.1	0.0613	3,249
	Testing	324874.6	0.0570	1,393
elastnet				
	Training	310358.5	0.0619	3,249
	Testing	324727.4	0.0574	1,393
sqlasso				
	Training	310176.3	0.0625	3,249
	Testing	324962.4	0.0567	1,393

Workflow

- Step 4: Evaluate prediction performance using testing sample

```
. lassogof ols lasso elastnet sqlasso, over(sample)
Penalized coefficients
```

Name	sample	MSE	R-squared	Obs
ols				
	Training	304368.1	0.0800	3,249
	Testing	328554.5	0.0463	1,393
lasso				
	Training	310573.1	0.0613	3,249
	Testing	324874.6	0.0570	1,393
elastnet				
	Training	310358.5	0.0619	3,249
	Testing	324727.4	0.0574	1,393
sqlasso				
	Training	310176.3	0.0625	3,249
	Testing	324962.4	0.0567	1,393

- Elastic-net is the best method (lowest MSE in the testing sample)

Workflow

- Step 5: Compute predictions using the best estimator

```
. quietly elasticnet linear bweight $covs, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnetfull
. use cattaneo2_new, clear
(New data)
. estimates restore elastnetfull
(results elastnetfull are active now)
. predict yhat_pen
(options xb penalized assumed; linear prediction with penalized coefficients)
. predict yhat_postsel, postselection
(option xb assumed; linear prediction with postselection coefficients)
```

Workflow

- Step 5: Compute predictions using the best estimator

```
. quietly elasticnet linear bweight $covs, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnetfull
. use cattaneo2_new, clear
(New data)
. estimates restore elastnetfull
(results elastnetfull are active now)
. predict yhat_pen
(options xb penalized assumed; linear prediction with penalized coefficients)
. predict yhat_postsel, postselection
(option xb assumed; linear prediction with postselection coefficients)
```

  - By default, **`predict`** uses the penalized coefficients

Workflow

- Step 5: Compute predictions using the best estimator

```
. quietly elasticnet linear bweight $covs, alpha(0.2 0.5 0.75 0.9)
. estimates store elastnetfull
. use cattaneo2_new, clear
(New data)
. estimates restore elastnetfull
(results elastnetfull are active now)
. predict yhat_pen
(options xb penalized assumed; linear prediction with penalized coefficients)
. predict yhat_postsel, postselection
(option xb assumed; linear prediction with postselection coefficients)
```

  - By default, **predict** uses the penalized coefficients
  - The **postselection** option uses post-selection coefficients (OLS on variables selected by **elasticnet**). They are expected to perform better in out-of-sample prediction than the penalized coefficients

Display **lasso** output

```
. estimates restore lasso
(results lasso are active now)

. lasso

Lasso linear model                          No. of obs         =      3,249
                                            No. of covariates  =        117
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     115.008         0       0.0003     330748.1
      26 |   lambda before    11.23639        17       0.0514     313825.7
    * 27 | selected lambda    10.23818        17       0.0515     313799.9
      28 |    lambda after     9.32865        19       0.0515     313823.9
      39 |     last lambda    3.352543        28       0.0501     314281.9
--------------------------------------------------------------------------
* lambda selected by cross-validation.
```

Display `lasso` output

```
. estimates restore lasso
(results lasso are active now)
. lasso
Lasso linear model                          No. of obs        =      3,249
                                            No. of covariates =        117
Selection: Cross-validation                 No. of CV folds   =         10
```

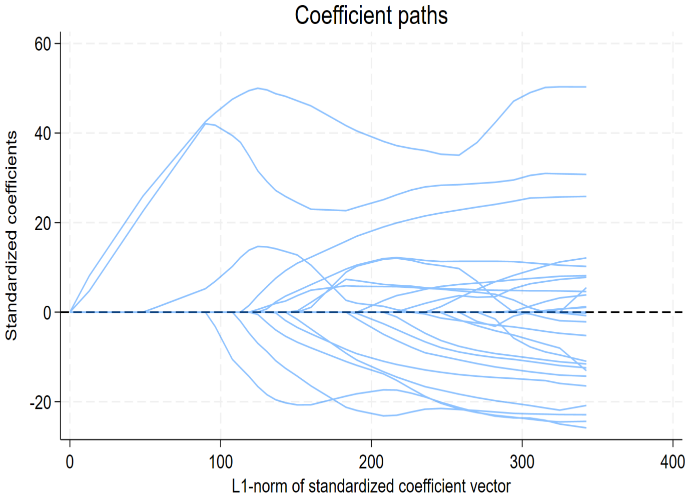
ID	Description	lambda	No. of nonzero coef.	Out-of-sample R-squared	CV mean prediction error
1	first lambda	115.008	0	0.0003	330748.1
26	lambda before	11.23639	17	0.0514	313825.7
* 27	selected lambda	10.23818	17	0.0515	313799.9
28	lambda after	9.32865	19	0.0515	313823.9
39	last lambda	3.352543	28	0.0501	314281.9

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

- Notice that the number of nonzero coefficients increases as λ decreases

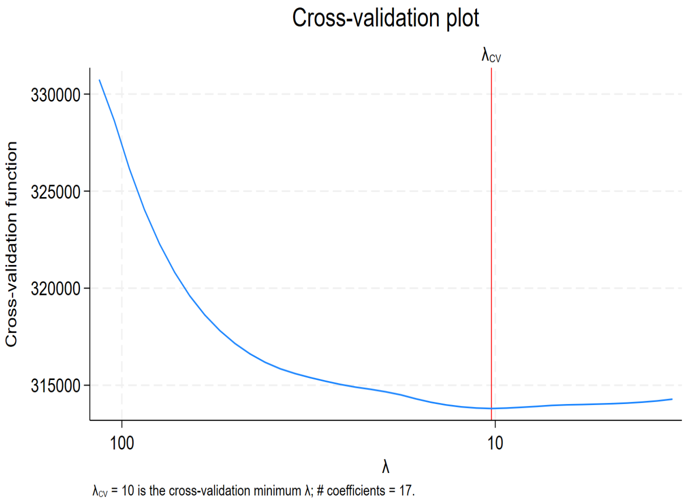
Plot path of coefficients after lasso

```
. coefpath
```



Plot cross-validation function after lasso

```
. cvplot
```



Display knot table (**lassoknots**)

```
. lassoknots

-------------------------------------------------------------------------------
       |                No. of   CV mean |
       |               nonzero     pred. |          Variables (A)dded, (R)emoved,
    ID |     lambda      coef.     error |               or left (U)nchanged
-------+---------------------------------+-------------------------------------
     2 |    104.791          2  328633.1 | A 0.msmoke#c.mage
       |                                 |   1.mmarried#c.fedu
     6 |   72.22835          3  320818.5 | A 0.msmoke#c.fedu
    11 |   45.36151          4  316613.1 | A 0.mmarried
(output omitted)
  * 27 |   10.23818         17  313799.9 | U
    28 |    9.32865         19  313823.9 | A 1.mhisp#c.medu
       |                                 |   2.msmoke#c.fage
    29 |   8.499919         18  313863.1 | R 0.msmoke#c.fedu
(output omitted)
    39 |   3.352543         28  314281.9 | A c.mage#c.monthslb
       |                                 |   0.prenatal1#c.mage
-------------------------------------------------------------------------------
* lambda selected by cross-validation.
```

Display knot table (`lassoknots`)

```
. lassoknots

-------------------------------------------------------------------------------------------
     |                  No. of   CV mean |
     |                 nonzero     pred. |       Variables (A)dded, (R)emoved,
  ID |     lambda        coef.     error |            or left (U)nchanged
-------------------------------------------------------------------------------------------
   2 |    104.791            2  328633.1 | A 0.msmoke#c.mage
     |                                   |   1.mmarried#c.fedu
   6 |   72.22835            3  320818.5 | A 0.msmoke#c.fedu
  11 |   45.36151            4  316613.1 | A 0.mmarried
(output omitted)
 * 27 |   10.23818           17  313799.9 | U
  28 |    9.32865           19  313823.9 | A 1.mhisp#c.medu
     |                                   |   2.msmoke#c.fage
  29 |   8.499919           18  313863.1 | R 0.msmoke#c.fedu
(output omitted)
  39 |   3.352543           28  314281.9 | A c.mage#c.monthslb
     |                                   |   0.prenatal1#c.mage
-------------------------------------------------------------------------------------------
* lambda selected by cross-validation.
```

- **`lassoselect`** can be used to pick a different λ value (sensitivity analysis)

Methods for selecting the value of λ

- Cross-validation (default) computes out-of-sample predictions MSEs using 10 folds and selects the λ with minimum MSE (**`selection(cv)`**)

Methods for selecting the value of λ

- Cross-validation (default) computes out-of-sample predictions MSEs using 10 folds and selects the λ with minimum MSE (**`selection(cv)`**)
- Adaptive lasso computes iterative 10-fold cross-validated lassos with larger penalty weights on small coefficients than a regular lasso. Covariates with large coefficients are more likely to be selected than covariates with small coefficients (**`selection(adaptive)`**)

Methods for selecting the value of λ

- Cross-validation (default) computes out-of-sample predictions MSEs using 10 folds and selects the λ with minimum MSE (**`selection(cv)`**)
- Adaptive lasso computes iterative 10-fold cross-validated lassos with larger penalty weights on small coefficients than a regular lasso. Covariates with large coefficients are more likely to be selected than covariates with small coefficients (**`selection(adaptive)`**)
- Plugin method uses the structure of the model and advanced theoretical results to find the smallest λ that dominates the noise, given estimates of the penalty weights (**`selection(plugin)`**)

Methods for selecting the value of λ

- Cross-validation (default) computes out-of-sample predictions MSEs using 10 folds and selects the λ with minimum MSE (**`selection(cv)`**)
- Adaptive lasso computes iterative 10-fold cross-validated lassos with larger penalty weights on small coefficients than a regular lasso. Covariates with large coefficients are more likely to be selected than covariates with small coefficients (**`selection(adaptive)`**)
- Plugin method uses the structure of the model and advanced theoretical results to find the smallest λ that dominates the noise, given estimates of the penalty weights (**`selection(plugin)`**)
- Bayesian information criteria (BIC) finds λ that minimizes the BIC statistic (**`selection(bic)`**, Stata 17)

Methods for selecting the value of λ

- Cross-validation (default) computes out-of-sample predictions MSEs using 10 folds and selects the λ with minimum MSE (**`selection(cv)`**)
- Adaptive lasso computes iterative 10-fold cross-validated lassos with larger penalty weights on small coefficients than a regular lasso. Covariates with large coefficients are more likely to be selected than covariates with small coefficients (**`selection(adaptive)`**)
- Plugin method uses the structure of the model and advanced theoretical results to find the smallest λ that dominates the noise, given estimates of the penalty weights (**`selection(plugin)`**)
- Bayesian information criteria (BIC) finds λ that minimizes the BIC statistic (**`selection(bic)`**, Stata 17)
- Manual selection (**`lassoselect`**)

Choosing λ using the **selection()** option

```
. quietly lasso linear bweight $covs
. estimates store cv
. quietly lasso linear bweight $covs, selection(adaptive)
. estimates store adaptive
. quietly lasso linear bweight $covs, selection(plugin)
. estimates store plugin
. quietly lasso linear bweight $covs, selection(bic)
. estimates store bic
```

Display basic information about lassos (**lassoinfo**)

```
. lassoinfo cv adaptive plugin bic
    Estimate: cv
     Command: lasso
```

Dependent variable	Model	Selection method	Selection criterion	lambda	No. of selected variables
bweight	linear	cv	CV min.	9.867787	19

```
    Estimate: adaptive
     Command: lasso
```

Dependent variable	Model	Selection method	Selection criterion	lambda	No. of selected variables
bweight	linear	adaptive	CV min.	3.64e+08	13

```
    Estimate: plugin
     Command: lasso
```

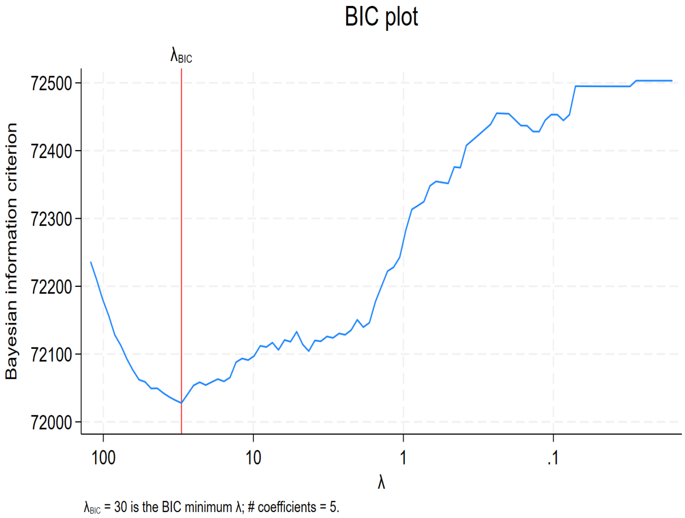
Dependent variable	Model	Selection method	lambda	No. of selected variables
bweight	linear	plugin	.0627659	6

```
    Estimate: bic
     Command: lasso
```

Dependent variable	Model	Selection method	Selection criterion	lambda	No. of selected variables
bweight	linear	bic	BIC min.	30.1348	5

Plot Bayesian information criterion function after lasso

```
. bicplot
```



Display coefficients after lasso (`lassocoef`)

```
. lassocoef cv adaptive plugin bic, display(coef, standardized)
```

	cv	adaptive	plugin	bic
mmarried				
Not married	-22.11483	-41.92165	-3.921652	-13.51986
Married	8.19e-10		3.38e-10	
msmoke				
0 daily	19.53755			
mmarried#c.fage				
Not married	-7.447833	-5.868928		
mmarried#c.medu				
Not married	-1.549706			
Married	16.17073	15.6786	21.32031	14.89484
mmarried#c.fedu				
Married	13.60921		16.50398	19.33894
foreign#c.fage				
1	-4.972952	-17.05522		
foreign#c.fedu				
0	3.90694			
alcohol#c.mage				
1	-3.167769	-6.859747		
msmoke#c.mage				
0 daily	52.73363	84.46094	57.27693	63.43016

(output omitted)

Lasso for inference

Motivation

- Ideally we would have a correct model for both data and theory. If so, we would just need to fit the model (using an appropriate estimator) and we report point estimates, standard errors, p-values, and confidence intervals

Motivation

- Ideally we would have a correct model for both data and theory. If so, we would just need to fit the model (using an appropriate estimator) and we report point estimates, standard errors, p-values, and confidence intervals
- In practice things are different. Consider the linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\alpha + \beta_0 + \mathbf{x}\beta'$$

    - We may fit many models with different subsets of controls

Motivation

- Ideally we would have a correct model for both data and theory. If so, we would just need to fit the model (using an appropriate estimator) and we report point estimates, standard errors, p-values, and confidence intervals
- In practice things are different. Consider the linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\alpha + \beta_0 + \mathbf{x}\beta'$$

  - We may fit many models with different subsets of controls
  - And, we would choose a model that we believe is the "best" to represent our theory or proposition. We apply an estimator and perform statistical inference

Motivation

- Ideally we would have a correct model for both data and theory. If so, we would just need to fit the model (using an appropriate estimator) and we report point estimates, standard errors, p-values, and confidence intervals
- In practice things are different. Consider the linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\alpha + \beta_0 + \mathbf{x}\beta'$$

  - We may fit many models with different subsets of controls
  - And, we would choose a model that we believe is the "best" to represent our theory or proposition. We apply an estimator and perform statistical inference
  - But, if we do not account for the model-selection process, inference would be invalid

Motivation

- Ideally we would have a correct model for both data and theory. If so, we would just need to fit the model (using an appropriate estimator) and we report point estimates, standard errors, p-values, and confidence intervals
- In practice things are different. Consider the linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\alpha + \beta_0 + \mathbf{x}\beta'$$

  - We may fit many models with different subsets of controls
  - And, we would choose a model that we believe is the "best" to represent our theory or proposition. We apply an estimator and perform statistical inference
  - But, if we do not account for the model-selection process, inference would be invalid
- Suppose there are many potential controls. Which controls should we include in the model? How to perform valid inference on the variables of interest?

Invalid approach

1. Apply lasso for y on the variables of interest (**d** vector) and the controls (**x** vector) forcing the variables of interest to be in the model. This selects a subset of controls (**x**$*$ vector)

Invalid approach

1. Apply lasso for y on the variables of interest (**d** vector) and the controls (**x** vector) forcing the variables of interest to be in the model. This selects a subset of controls ($\mathbf{x}*$ vector)
2. Fit OLS regression for y on the variables of interest and the selected controls

Invalid approach

1. Apply lasso for y on the variables of interest (**d** vector) and the controls (**x** vector) forcing the variables of interest to be in the model. This selects a subset of controls ($\mathbf{x}*$ vector)
2. Fit OLS regression for y on the variables of interest and the selected controls
3. Perform inference on the coefficients for the variables of interest (parameter vector α)

Invalid approach

1. Apply lasso for y on the variables of interest (**d** vector) and the controls (**x** vector) forcing the variables of interest to be in the model. This selects a subset of controls (**x**$*$ vector)
2. Fit OLS regression for y on the variables of interest and the selected controls
3. Perform inference on the coefficients for the variables of interest (parameter vector α)

- This approach would produce invalid statistical inference. Why?

Invalid approach

1. Apply lasso for y on the variables of interest (**d** vector) and the controls (**x** vector) forcing the variables of interest to be in the model. This selects a subset of controls ($\mathbf{x}*$ vector)
2. Fit OLS regression for y on the variables of interest and the selected controls
3. Perform inference on the coefficients for the variables of interest (parameter vector α)

- This approach would produce invalid statistical inference. Why?
  - Model-selection techniques inevitably make mistakes selecting controls

Invalid approach

1. Apply lasso for y on the variables of interest (**d** vector) and the controls (**x** vector) forcing the variables of interest to be in the model. This selects a subset of controls ($\mathbf{x}*$ vector)
2. Fit OLS regression for y on the variables of interest and the selected controls
3. Perform inference on the coefficients for the variables of interest (parameter vector α)

- This approach would produce invalid statistical inference. Why?
    - Model-selection techniques inevitably make mistakes selecting controls
    - The actual sampling distribution of α is not concentrated (multiple modes). (Leeb and Pötscher, 2005)

Solutions

- Double selection: Belloni et al. (2014), Belloni et al. (2016) (**`dsregress`**, **`dslogit`**, and **`dspoisson`**)

Solutions

- Double selection: Belloni et al. (2014), Belloni et al. (2016) (**`dsregress`**, **`dslogit`**, and **`dspoisson`**)
- Partialing-out: Belloni et al. (2016), Chernozhukov et al. (2015) (**`poregress`**, **`poivregress`**, **`pologit`**, and **`popoisson`**)

Solutions

- Double selection: Belloni et al. (2014), Belloni et al. (2016) (**`dsregress`**, **`dslogit`**, and **`dspoisson`**)
- Partialing-out: Belloni et al. (2016), Chernozhukov et al. (2015) (**`poregress`**, **`poivregress`**, **`pologit`**, and **`popoisson`**)
- Cross-fit partialing-out (double machine learning): Chernozhukov et al. (2018) (**`xporegress`**, **`xpoivregress`**, **`xpologit`**, and **`xpopoisson`**)

Solutions

- Double selection: Belloni et al. (2014), Belloni et al. (2016) (**`dsregress`**, **`dslogit`**, and **`dspoisson`**)
- Partialing-out: Belloni et al. (2016), Chernozhukov et al. (2015) (**`poregress`**, **`poivregress`**, **`pologit`**, and **`popoisson`**)
- Cross-fit partialing-out (double machine learning): Chernozhukov et al. (2018) (**`xporegress`**, **`xpoivregress`**, **`xpologit`**, and **`xpopoisson`**)
- "These solutions all use multiple lassos and moment conditions that are robust to the model-selection mistakes that lasso makes"

Solutions

- Double selection: Belloni et al. (2014), Belloni et al. (2016) (**`dsregress`**, **`dslogit`**, and **`dspoisson`**)
- Partialing-out: Belloni et al. (2016), Chernozhukov et al. (2015) (**`poregress`**, **`poivregress`**, **`pologit`**, and **`popoisson`**)
- Cross-fit partialing-out (double machine learning): Chernozhukov et al. (2018) (**`xporegress`**, **`xpoivregress`**, **`xpologit`**, and **`xpopoisson`**)
- "These solutions all use multiple lassos and moment conditions that are robust to the model-selection mistakes that lasso makes"
- By default, all of the command above fit the lassos using **`selection(plugin)`**

Example on lasso for inference with a linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\boldsymbol{\alpha} + \beta_0 + \mathbf{x}\boldsymbol{\beta}'$$

- y = wage (monthly wages)

Example on lasso for inference with a linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\boldsymbol{\alpha} + \beta_0 + \mathbf{x}\boldsymbol{\beta}'$$

- y = wage (monthly wages)
- $\mathbf{d}$ = (educ, tenure)
  - educ: Years of education
  - tenure: Years with current employer

Example on lasso for inference with a linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\boldsymbol{\alpha} + \beta_0 + \mathbf{x}\boldsymbol{\beta}'$$

- y = wage (monthly wages)
- $\mathbf{d}$ = (educ, tenure)
  - educ: Years of education
  - tenure: Years with current employer
- $\mathbf{x}$: vector of potential control variables
  - 6 continuous variables, 1 categorical variable, 5 binary variables
  - All main effects and all possible interactions generate 230 controls

Example on lasso for inference with a linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\boldsymbol{\alpha} + \beta_0 + \mathbf{x}\boldsymbol{\beta}'$$

- y = wage (monthly wages)
- **d** = (educ, tenure)
  - educ: Years of education
  - tenure: Years with current employer
- **x**: vector of potential control variables
  - 6 continuous variables, 1 categorical variable, 5 binary variables
  - All main effects and all possible interactions generate 230 controls
- Number of observations: 722

Example on lasso for inference with a linear model

$$\mathbf{E}[y|\mathbf{d},\mathbf{x}] = \mathbf{d}\boldsymbol{\alpha} + \beta_0 + \mathbf{x}\boldsymbol{\beta}'$$

- y = wage (monthly wages)
- **d** = (educ, tenure)
  - educ: Years of education
  - tenure: Years with current employer
- **x**: vector of potential control variables
  - 6 continuous variables, 1 categorical variable, 5 binary variables
  - All main effects and all possible interactions generate 230 controls
- Number of observations: 722
- Which controls should we include in the model to perform valid inference on α?

dsregress – Double-selection lasso linear regression

```
. use nlsy80

. global controls c.(meduc feduc sibs age iq kww)##(exper ///
>         pcollege married black south urban)

. dsregress wage educ tenure, controls($controls)

Estimating lasso for wage using plugin
Estimating lasso for educ using plugin
Estimating lasso for tenure using plugin

Double-selection linear model          Number of obs               =        722
                                       Number of controls          =        230
                                       Number of selected controls =         12
                                       Wald chi2(2)                =      17.03
                                       Prob > chi2                 =     0.0002

------------------------------------------------------------------------------
             |               Robust
        wage | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        educ |   29.29732    7.58747     3.86   0.000     14.42615    44.16849
      tenure |   5.105178   2.950394     1.73   0.084     -.677488    10.88784
------------------------------------------------------------------------------
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 ds_plugin
```

dsregress – Double-selection lasso linear regression

```
. use nlsy80
. global controls c.(meduc feduc sibs age iq kww)##(exper ///
>         pcollege married black south urban)
. dsregress wage educ tenure, controls($controls)
Estimating lasso for wage using plugin
Estimating lasso for educ using plugin
Estimating lasso for tenure using plugin
Double-selection linear model          Number of obs               =        722
                                       Number of controls          =        230
                                       Number of selected controls =         12
                                       Wald chi2(2)                =      17.03
                                       Prob > chi2                 =     0.0002
------------------------------------------------------------------------------
             |               Robust
        wage | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        educ |   29.29732    7.58747     3.86   0.000     14.42615    44.16849
      tenure |   5.105178   2.950394     1.73   0.084     -.677488    10.88784
------------------------------------------------------------------------------
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 ds_plugin
```

- Inference on controls would not be valid; and so, they are not reported

poregress – Partialing-out lasso linear regression

```
. poregress wage educ tenure, controls($controls)
Estimating lasso for wage using plugin
Estimating lasso for educ using plugin
Estimating lasso for tenure using plugin
Partialing-out linear model           Number of obs                =        722
                                      Number of controls           =        230
                                      Number of selected controls  =         12
                                      Wald chi2(2)                 =      17.77
                                      Prob > chi2                  =     0.0001

------------------------------------------------------------------------------
             |               Robust
        wage | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        educ |   29.61345   7.455942     3.97   0.000     15.00007    44.22683
      tenure |   4.995759   2.874894     1.74   0.082     -.63893     10.63045
------------------------------------------------------------------------------
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.
```

xporegress – Cross-fit partialing-out lasso linear regression

```
. xporegress wage educ tenure, controls($controls)

(output omitted)

Cross-fit partialing-out               Number of obs                  =        722
linear model                           Number of controls             =        230
                                       Number of selected controls    =         25
                                       Number of folds in cross-fit   =         10
                                       Number of resamples            =          1
                                       Wald chi2(2)                   =      18.00
                                       Prob > chi2                    =     0.0001

------------------------------------------------------------------------------
             |               Robust
        wage | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        educ |   29.62034   7.430891     3.99   0.000     15.05606    44.18462
      tenure |   5.082955   2.808093     1.81   0.070    -.420805    10.58672
------------------------------------------------------------------------------
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 after **xporegress**

```
. lassoinfo

    Estimate: active
     Command: xporegress
```

Variable	Model	Selection method	No. of selected variables: min	median	max
educ	linear	plugin	5	7	9
tenure	linear	plugin	1	2	3
wage	linear	plugin	4	6	8

lassoinfo after xporegress

```
. lassoinfo

    Estimate: active
     Command: xporegress

            |                        No. of selected variables
            |            Selection
   Variable |      Model     method      min    median      max
------------+---------------------------------------------------
       educ |     linear     plugin        5         7        9
     tenure |     linear     plugin        1         2        3
       wage |     linear     plugin        4         6        8
```

- By default, **lassoinfo** displays summary of lassos by variable

`lassoinfo` after `xporegress`

```
. lassoinfo

    Estimate: active
     Command: xporegress
```

Variable	Model	Selection method	No. of selected variables: min	median	max
educ	linear	plugin	5	7	9
tenure	linear	plugin	1	2	3
wage	linear	plugin	4	6	8

- By default, **`lassoinfo`** displays summary of lassos by variable
- The option **`each`** would display information of each lasso

General advice

- The cross-fit partialing-out estimators are the best ones (**`xporegress`**, **`xpologit`**, **`xpopoisson`**, **`xpoivregress`**). But, computations may take extremely long time

General advice

- The cross-fit partialing-out estimators are the best ones (**`xporegress`**, **`xpologit`**, **`xpopoisson`**, **`xpoivregress`**). But, computations may take extremely long time
- If you do not have the time, use either the partialing-out estimator (**`poregress`**, **`pologit`**, **`popoisson`**, **`poivregress`**) or the double-selection estimator (**`dsregress`**, **`dslogit`**, **`dspoisson`**)

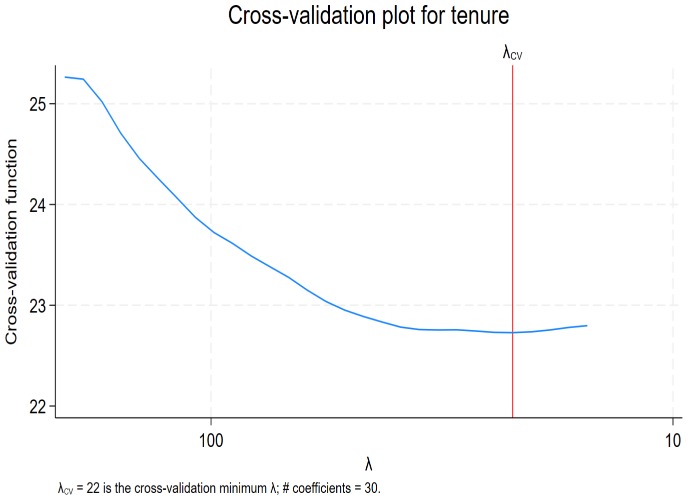
Customize individual lassos

```
. dsregress wage educ tenure,controls($controls) ///
>   lasso(wage,selection(adaptive)) ///
>   lasso(educ,selection(bic)) ///
>   sqrtlasso(tenure,selection(cv))
Estimating lasso for wage using adaptive
Estimating lasso for educ using BIC
Estimating square-root lasso for tenure using cv
Double-selection linear model          Number of obs               =        722
                                       Number of controls          =        230
                                       Number of selected controls =         54
                                       Wald chi2(2)                =      18.28
                                       Prob > chi2                 =     0.0001

------------------------------------------------------------------------------
             |               Robust
        wage | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
        educ |   34.66224   8.708087     3.98   0.000      17.5947    51.72978
      tenure |   4.881471   3.095039     1.58   0.115    -1.184694    10.94764
------------------------------------------------------------------------------
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.
```

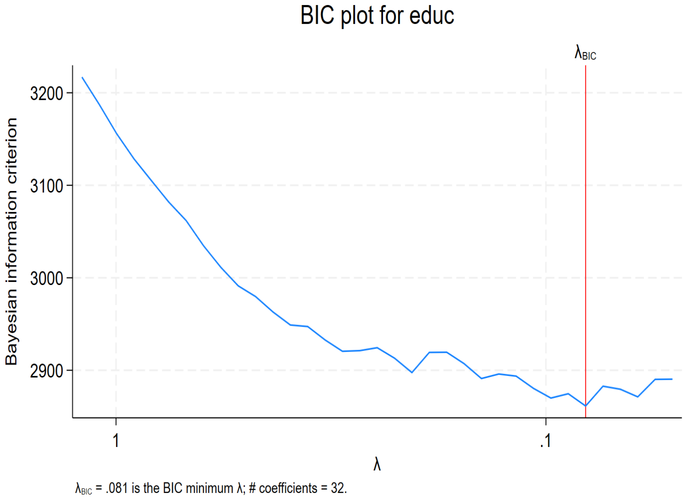
`cvplot` for a particular lasso

```
. cvplot, for(tenure)
```



`bicplot` for a particular lasso

```
.  bicplot, for(educ)
```



Other options for selecting controls

```
. quietly dsregress wage educ tenure, controls($controls) selection(cv)
. estimates store ds_cv
. quietly dsregress wage educ tenure, controls($controls) selection(adaptive)
. estimates store ds_adapt
. quietly dsregress wage educ tenure, controls($controls) selection(bic)
. estimates store ds_bic
. estimates table ds_plugin ds_cv ds_adapt ds_bic, b(%9.4f) se(%9.4f) p(%9.4f)
```

Variable	ds_plugin	ds_cv	ds_adapt	ds_bic
educ	29.2973	32.8323	34.1067	33.3164
	7.5875	8.8374	8.6690	8.6672
	0.0001	0.0002	0.0001	0.0001
tenure	5.1052	5.1631	4.8216	4.6522
	2.9504	3.0597	3.0784	3.0064
	0.0836	0.0915	0.1173	0.1218

Legend: b/se/p

Conclusion

Conclusion

- Lasso for prediction and model selection
  - We used **`lasso linear`**, **`elasticnet linear`**, and **`sqrtlasso`**. And, we used **`splitsample`** to help in choosing the best

Conclusion

- Lasso for prediction and model selection
  - We used **`lasso linear`**, **`elasticnet linear`**, and **`sqrtlasso`**. And, we used **`splitsample`** to help in choosing the best
  - Also available for **`lasso`** and **`elasticnet`**, there are sub-commands for logit, probit, Poisson, and Cox models

Conclusion

- Lasso for prediction and model selection
  - We used **`lasso linear`**, **`elasticnet linear`**, and **`sqrtlasso`**. And, we used **`splitsample`** to help in choosing the best
  - Also available for **`lasso`** and **`elasticnet`**, there are sub-commands for logit, probit, Poisson, and Cox models
- Lasso estimators for inference
  - We used **`dsregress`**, **`poregress`**, and **`xporegress`**.

Conclusion

- Lasso for prediction and model selection
  - We used **`lasso linear`**, **`elasticnet linear`**, and **`sqrtlasso`**. And, we used **`splitsample`** to help in choosing the best
  - Also available for **`lasso`** and **`elasticnet`**, there are sub-commands for logit, probit, Poisson, and Cox models
- Lasso estimators for inference
  - We used **`dsregress`**, **`poregress`**, and **`xporegress`**.
  - Also available are **`dslogit`**, **`dspoisson`**, **`poivregress`**, **`pologit`**, **`popoisson`**, **`xpoivregress`**, **`xpologit`**, **`xpopoisson`**, **`telasso`**

Conclusion

- Lasso for prediction and model selection
  - We used **`lasso linear`**, **`elasticnet linear`**, and **`sqrtlasso`**. And, we used **`splitsample`** to help in choosing the best
  - Also available for **`lasso`** and **`elasticnet`**, there are sub-commands for logit, probit, Poisson, and Cox models
- Lasso estimators for inference
  - We used **`dsregress`**, **`poregress`**, and **`xporegress`**.
  - Also available are **`dslogit`**, **`dspoisson`**, **`poivregress`**, **`pologit`**, **`popoisson`**, **`xpoivregress`**, **`xpologit`**, **`xpopoisson`**, **`telasso`**
  - Using lasso for prediction and listing the selected variables in estimation commands will generally lead to invalid statistical inference. Instead, use lasso inferential commands

Conclusion

- Lasso for prediction and model selection
  - We used **lasso linear**, **elasticnet linear**, and **sqrtlasso**. And, we used **splitsample** to help in choosing the best
  - Also available for **lasso** and **elasticnet**, there are sub-commands for logit, probit, Poisson, and Cox models
- Lasso estimators for inference
  - We used **dsregress**, **poregress**, and **xporegress**.
  - Also available are **dslogit**, **dspoisson**, **poivregress**, **pologit**, **popoisson**, **xpoivregress**, **xpologit**, **xpopoisson**, **telasso**
  - Using lasso for prediction and listing the selected variables in estimation commands will generally lead to invalid statistical inference. Instead, use lasso inferential commands
  - Use cross-fit partialing-out estimators if you have the time; otherwise, use either the partialing-out estimator or the double-selection estimator

References

- Belloni, A., V. Chernozhukov, and C. Hansen. 2014. Inference on treatment effects after selection among high-dimensional controls. The Review of Economic Studies 81(2): 608-650.
- Belloni, A., V. Chernozhukov, and Y. Wei. 2016. Post-selection inference for generalized linear models with many controls. Journal of Business & Economic Statistics 34(4): 606-619.
- Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. 2018. Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal 21(1): C1-C68.
- Chernozhukov, V., C. Hansen, and M. Spindler. 2015. Post-selection and post-regularization inference in linear models with many controls and instruments. American Economic Review 105(5): 486-90.
- Leeb, H., and B. M. Pötscher. 2005. Model selection and inference: Facts and fiction. Econometric Theory 21(1): 21-59.
- Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58(1): 267-288.