Description

Lasso was originally an acronym for "least absolute shrinkage and selection operator". Today, lasso is considered a word and not an acronym.

Lasso is used for prediction, for model selection, and as a component of estimators to perform inference.

Lasso, elastic net, and square-root lasso are designed for model selection and prediction. Stata's `lasso`, `elasticnet`, and `sqrtlasso` commands implement these methods. `lasso` and `elasticnet` fit continuous, binary, count, and failure-time outcomes, while `sqrtlasso` fits continuous outcomes.

Stata also provides lasso commands for inference. They use lassos to select control variables that appear in the model, and they estimate coefficients and standard errors for a specified subset of covariates.

Stata's lasso inference commands implement methods known as double selection, partialing out, and cross-fit partialing out. With each of these methods, linear, logistic, or Poisson regression can be used to model a continuous, binary, or count outcome. Partialing out and cross-fit partialing out also allow for endogenous covariates in linear models.

Stata also provides a specialized lasso inference command for estimating treatment effects while using lassos to select control variables. `telasso` estimates the average treatment effect (ATE), average treatment effect on the treated (ATET), or potential-outcome means (POMs); see [CAUSAL] **telasso**.

This entry provides an overview of lasso for prediction, model selection, and inference and an introduction to Stata's suite of lasso commands.

Remarks and examples

Remarks are presented under the following headings:

Summary of Stata's lasso and elastic-net features

For those of you who are already experts on lasso, here is an overview full of buzz words without explanations.

Stata provides three estimation commands for prediction and model selection.

lasso   fits linear, logit, probit, Poisson, and Cox proportional hazards models. The final model is selected using cross-validation (CV), adaptive lasso, plugin estimators, or the Bayesian information criterion (BIC) function.

elasticnet   also fits linear, logit, probit, Poisson, and Cox proportional hazards models. `elasticnet` uses CV or the BIC function to select models.

sqrtlasso   fits square-root lassos for linear models. The final model is selected using CV, plugin estimators, or the BIC function.

After fitting a model, you obtain out-of-sample predictions by loading another dataset and typing

    . predict *newvarname*

Stata provides another 11 lasso commands for use in inference. These 11 commands are arranged in three groups.

ds commands   perform double-selection lasso:
                      dsregress for linear models,
                      dslogit for logit models, and
                      dspoisson for Poisson models.

po commands   perform partialing-out lasso:
                      poregress for linear models,
                      pologit for logit models,
                      popoisson for Poisson models, and
                      poivregress for linear models with endogenous covariates.

xpo commands   perform cross-fit partialing-out lasso, also known as double machine learning:
                      xporegress for linear models,
                      xpologit for logit models,
                      xpopoisson for Poisson models, and
                      xpoivregress for linear models with endogenous covariates.

Stata provides one additional lasso command for use in inference when your objective is to estimate treatment effects.

telasso   fits linear, logit, probit, and Poisson models and estimates the ATE, ATET, or POMs.

Stata provides two preestimation commands that help you prepare your data and specify covariates.

splitsample   divides your sample into k random subsamples. Use it for producing
                   subsamples for training, validation, and prediction.

vl   creates named lists of variables to be included in lasso. Sometimes,
                   you will want to specify many potential covariates without typing every
                   variable name. vl creates named variable lists that can be used as
                   command arguments. Lists can contain hundreds or thousands of
                   variable names.

Stata provides eight postestimation commands that help you evaluate the selected model:

bicplot   graphs the BIC function.

cvplot   graphs the CV function.

coefpath   graphs coefficient paths.

lassoknots   displays a knot table for covariates as they enter or leave the model and
                   measures of fit.

lassogof   reports fit statistics that help you evaluate the predictive ability of a model.
                   It does this for one model or for multiple models in the same table.

lassocoef   lists the selected variables in the model. It does this for one model
                   or for multiple models in the same table.

lassoselect   selects a different model from the one chosen by the estimation command.

lassoinfo   reports lasso information such as the dependent variable, selection method,
                   and number of nonzero coefficients for one or more models.

What is lasso?

Lasso is a method for selecting and fitting covariates that appear in a model. The lasso command can fit linear, logit, probit, Poisson, and Cox proportional hazards models. Let's consider a linear model, a model of y on x1, x2, …, xp. You would ordinarily fit this model by typing

        . regress y x1 x2 ... xp

Now assume that you are uncertain which variables (covariates) belong in the model, although you are certain that some of them do and the number of them is small relative to the number of observations in your dataset, N. In that case, you can type

        . lasso linear y x1 x2 ... xp

You can specify hundreds or even thousands of covariates. You can even specify more covariates than there are observations in your data! The covariates you specify are the potential covariates from which lasso selects.

Lasso is used in three ways:

1. Lasso is used for prediction.
2. Lasso is used for model selection.
3. Lasso is used for inference.

By prediction, we mean predicting the value of an outcome conditional on a large set of potential regressors. And we mean predicting the outcome both in and out of sample.

By model selection, we mean selecting a set of variables that predicts the outcome well. We do not mean selecting variables in the true model or placing a scientific interpretation on the coefficients. Instead, we mean selecting variables that correlate well with the outcome in one dataset and testing whether those same variables predict the outcome well in other datasets.

By inference, we mean inference for interpreting and giving meaning to the coefficients of the fitted model. Inference is concerned with estimating effects of variables in the true model and estimating standard errors, confidence intervals, p-values, and the like.

Lasso for prediction

Lasso was invented by Tibshirani (1996) and has been commonly used in building models for prediction. Hastie, Tibshirani, and Wainwright (2015) provide an excellent introduction to the mechanics of the lasso and to the lasso as a tool for prediction. See Bühlmann and van de Geer (2011) for more technical discussion and clear discussion of the properties of lasso under different assumptions. See Cameron and Trivedi (2022, chap. 28) for an introduction to lasso for prediction and for inference with examples using Stata.

Lasso does not necessarily select the covariates that appear in the true model, but it does select a set of variables that are correlated with them. If lasso selects potential covariate x47, that means x47 belongs in the model or is correlated with variables that belong in the model. If lasso omits potential covariate x52, that means x52 does not belong in the model or belongs but is correlated with covariates that were already selected. Because we are interested only in prediction, we are not concerned with the exact variables selected, only that they are useful for prediction.

The model lasso selects is suitable for making predictions in samples outside the one you used for estimation. Everyone knows about the danger of overfitting. Fit a model on one set of data and include too many variables, and the result will exploit features randomly unique to those data that will not be replicated in other data.

"Oh," you may be thinking, "you mean that I can split my data into an estimation sample and a hold-out sample, and after fitting, I can evaluate the model in the hold-out sample." That is not what we mean, although you can do this, and it is sometimes a good idea to do so. We mean that lasso works to avoid the problem of overfitting by minimizing an estimate of the out-of-sample prediction error.

How lasso for prediction works

Lasso finds a solution for

$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \cdots + \beta_p \mathbf{x}_p + \boldsymbol{\epsilon}$$

by minimizing the prediction error subject to the constraint that the model is not too complex—that is, it is sparse. Lasso measures complexity by the sum of the absolute values of $\beta_1, \beta_2, \ldots, \beta_p$. The solution is obtained by minimizing

$$\frac{1}{2N}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}')'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}') + \lambda \sum_{j=1}^{p} |\beta_j| \tag{1}$$

The first term, $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}')'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}')$, is the in-sample prediction error. It is the same value that least squares minimizes.

The second term, $\lambda \sum_j |\beta_j|$, is a penalty that increases in value the more complex the model. It is this term that causes lasso to omit variables. They are omitted because of the nondifferentiable kinks in the $\sum_j |\beta_j|$ absolute value terms. Had the kinks not been present—think of squared complexity terms rather than absolute value—none of the coefficients would be exactly zero. The kinks cause some coefficients to be zero.

If you minimized (1) with respect to the β_j's and λ, the solution would be $\lambda = 0$. That would set the penalty to zero. $\lambda = 0$ corresponds to a model with maximum complexity.

Lasso proceeds differently. It minimizes (1) for given values of λ. Lasso then chooses one of those solutions as best based on another criterion, such as an estimate of the out-of-sample prediction error.

When we use lasso for prediction, we must assume the unknown true model contains few variables relative to the number of observations, N. This is known as the sparsity assumption. How many true variables are allowed for a given N? We can tell you that the number cannot be greater than something proportional to $\sqrt{N}/\ln q$, where $q = \max\{N, p\}$ and p is the number of potential variables. We cannot, however, say what the constant of proportionality is. That this upper bound decreases with q can be viewed as the cost of performing covariate selection.

Lasso provides various ways of selecting λ: through CV, adaptive lasso, a plugin estimator, or minimizing the Bayesian information criterion (BIC) function. CV selects the λ that minimizes an estimate of the out-of-sample prediction error. Adaptive lasso performs multiple lassos, each with CV. After each lasso, variables with zero coefficients are removed, and remaining variables are given penalty weights designed to drive small coefficients to zero. Thus, adaptive lasso typically selects fewer covariates than CV.

The plugin method was designed to achieve an optimal sparsity rate. It tends to select a larger λ than CV and, therefore, fewer covariates in the final model. The number of covariates selected by minimizing BIC typically lies between the number selected by CV and the number selected by the plugin method; however, BIC tends to be more similar to the number selected by the plugin method. Furthermore, BIC does not require a complex derivation as does the plugin, so like CV, it can be applied in a more general context. See [LASSO] **lasso** and [LASSO] **lasso fitting** for more information on the methods of selecting λ, their differences, and how you can control the selection process.

Stata commands for prediction

We described the linear lasso model in the last section, but the concepts we have discussed apply to models for binary and count outcomes as well.

To fit a linear lasso model, we might type

```
. lasso linear y x1-x500
```

and `lasso` will select a subset of variables from `x1` to `x500` that can be used in prediction.

If we have a binary outcome, we could instead fit a logit model by typing

```
. lasso logit y x1-x500
```

or a probit model by typing

```
. lasso probit y x1-x500
```

For a count outcome, we could fit a Poisson model by typing

```
. lasso poisson y x1-x500
```

For failure-time data that has been `stset`, we could fit a Cox proportional hazards model by typing

```
. lasso cox x1-x500
```

After any of these `lasso` commands, we can use `predict` to obtain predictions of y.

For examples demonstrating how to use the `lasso` command to fit models suitable for prediction, see *Remarks and examples* in [LASSO] **lasso** and also see [LASSO] **lasso examples**.

Stata also has commands for fitting elastic nets and square-root lassos for prediction. See [LASSO] **elasticnet** and [LASSO] **sqrtlasso** for more information and examples.

Lasso for model selection

Model selection is an overloaded term that implies different things in different disciplines. To some, it implies finding a true model or data-generating process. To some, it implies less. Here model selection means finding a model that fits the data, not finding a model that allows for interpreting estimated coefficients as effects. If this is your interest, see *Lasso for inference* below.

Lasso for model selection builds on lasso for prediction. In fact, the same lasso methods are used in both cases. However, the goal of the analysis is different.

Model selection uses lasso to select variables in one dataset and then fits models using the selected variables in other datasets. For example, consider finding genes correlated with an outcome in microarray data. One approach starts with lasso. Researchers use it as a sieve to select the important predictors. They go on to test whether those predictors (genetic markers) work in other datasets. Note that these researchers are not giving scientific meaning to the estimated coefficients. They are looking only for markers that correlate well with an outcome.

We can perform these types of tests because our interest lies in the selected model rather than the true coefficients of the data-generating process (DGP), sometimes called the data-generating mechanism (DGM). Interpretation is conditional on the selected model and cannot be interpreted as causal. See, for instance, Lee et al. (2016). As Berk et al. (2013) put it, the goal is "... to merely describe association between predictor and response variables; no data generating or causal claims are implied."

Lasso for inference

When we use lasso for inference, we are interested in interpreting the estimated coefficients. We are also interested in standard errors, hypothesis tests, confidence intervals, comparisons across levels, and the like. We want to interpret the results in the same way we interpret results from standard regression models.

Why do we need special lasso methods for inference?

It may be tempting to use lasso to select covariates and then use `regress` (or `logit`, `probit`, or `poisson`) to fit a model with the selected covariates. The results from the regression provide estimated coefficients and standard errors, confidence intervals, and p-values.

This approach does not work. Why?

Consider fitting a classic regression model. The standard error for a coefficient tells us about the distribution of the coefficient in repeated sampling. The 95% confidence interval includes the true value of the coefficient in 95% of repeated samples. Although we do not actually have repeated samples from the population, the standard errors allow us to account for sample-to-sample variability when making inferences.

If we use lasso to select covariates and then use `regress` to fit a model with only the selected covariates, the results will be problematic for use in inference for a few reasons.

First, when we use lasso, or any variable-selection method, we introduce a new source of variability. If we actually drew repeated samples from a population and used `lasso` to select covariates on each one, different covariates would be selected in each dataset. However, we have selected covariates using only a single sample. The standard errors reported by `regress` do not account for the sample-to-sample variability in the variable selection.

Second, lasso tends to omit covariates with small coefficients. This problem arises because lasso minimizes prediction error subject to the constraint that the model is not too complex, and lasso measures complexity by the sum of the absolute values of the coefficients. Covariates with small coefficients tend to be entrapped by the constraint. Small coefficients of covariates that belong in the model look just like small coefficients of variables that do not. Mistakenly omitted covariates, even those with small coefficients, can bias other coefficients. That bias is not solely a function of the size of the coefficient. See, for instance, Leeb and Pötscher (2005, 2006, 2008) and Belloni, Chernozhukov, and Hansen (2014).

And then there are more mundane reasons the selected variables can differ from the true variables. Imagine that you have fit a lasso model. You look at the results and observe that region-of-country covariates are included. You are surprised because you can think of no reason why they should be and wonder whether you are about to make an interesting discovery. You look at the lasso results in hopes of finding an explanation. You discover that income was excluded despite your expectations to the contrary. You find that age and education were included, but that does not surprise you. But region, age, and education are predictors of income. That could be the entire reason why region covariates were included. They were included only because income was excluded. Or it could be something deeper.

In general, the variables selected by lasso do not even converge to the ones in the true model as the number of observations goes to infinity. Lasso tends to omit covariates that have small coefficients in favor of irrelevant ones (variables not in the true model) that are correlated with the error term.

For these reasons, we must use lasso-based methods that are designed specifically for inference when we want to interpret coefficients.

Methods of lasso for inference

With lasso inferential methods, researchers wish to interpret the covariates selected by lasso in the context of the DGP. They apply causal interpretations to the results. This approach accounts for the fact that lasso does not select the true model with probability one, and it accounts for the errors that arise in model selection. To achieve this DGP causal interpretation, you must perform the selection process with

resampling. Thus, more than split sampling is needed to obtain consistent standard errors. See Belloni, Chernozhukov, and Hansen (2014) for an excellent introduction to using lasso to perform inference and make causal interpretations.

When your interest is in the underlying DGP, there are various ways of using lasso to estimate the effects of a few covariates that you have chosen a priori. These methods may be used when you know there are more covariates in the model and that they are sparse (relative to N). These methods apply, for instance, to performing inference about the effect of smoking on a health outcome when you know that lots of other variables potentially affect the outcome but do not know which ones.

The double-selection, partialing-out, and cross-fit partialing-out lassos provided by Stata can handle such problems.

Other methods for inference have been discussed in the literature. For instance, see van de Geer et al. (2014), Javanmard and Montanari (2014), and Zhang and Zhang (2014). The methods developed there are not implemented in Stata. While they have some appealing theoretical properties, they have not yet been much used in applied work.

Stata commands for inference

For inference, multistage extensions of lasso provide standard errors for a subset of variables that you specify. Imagine that you wish to estimate the coefficients for d1 and d2 in the model that includes other covariates:

```
. regress y d1 d2 x1-x500
```

Covariates x1 through x500 are control variables, some of which you need to include to obtain valid results for d1 and d2. Suppose your data contain 1,000 observations.

If all 500 covariates belong in the model, there is no way to proceed. Get more data. If only a small subset of them is required and you simply do not know which they are, there is a lasso-based solution. Type

```
. dsregress y d1 d2, controls(x1-x500)
```

Coefficients and standard errors for d1 and d2 will be reported. dsregress will use lasso to select from the 500 covariates and do that in a way that is robust to the model-selection mistakes that lasso makes because of sampling variability. There is no a priori limit on the number of **d** variables you can specify. But more variables mean more computation time. Time is roughly proportional to the number of **d** variables.

Stata provides three methods to fit these types of inferential models. They are

1. the ds double-selection commands: dsregress, dslogit, and dspoisson.

2. the po partialing-out commands: poregress, pologit, popoisson, and poivregress.

3. the xpo cross-fit partialing-out commands, also known as double machine learning: xporegress, xpologit, xpopoisson, and xpoivregress.

All three methods require a sparsity assumption. As with lasso for prediction, these methods of lasso for inference rely on the assumption that the number of nonzero coefficients in the true model is small relative to the number of observations and that the coefficients are large enough relative to error variance to be selected by the lasso.

ds and po are asymptotically equivalent. `poivregress` can handle d1 and d2 being endogenous in linear models. It does this using instrumental variables. You specify a set of potential instruments, and lasso will select from among them. You can have many potential control variables and many potential instruments; the number of each can be greater than N.

xpo is the most computationally intensive of the three methods. It is also generally viewed as superior to ds and po because it allows a weaker definition of sparsity. The sparsity bound for the ds and po methods grows in proportion to $\sqrt{N}$, while the sparsity bound for the xpo method grows in proportion to N.

For information on the assumptions and how the ds, po, and xpo commands work, see [LASSO] **Lasso inference intro** and [LASSO] **Inference requirements**.

For examples of fitting lasso inferential models, see [LASSO] **Inference examples**.

Where to learn more

After reading this intro, you may want to learn more about lasso for prediction and model selection, lasso for inference, and syntax for lasso commands, or you may just want to see some examples. Here we provide a guide to the entries in this manual that you may want to read next.

If you are interested in lasso for prediction or model selection, you may want to go directly to the syntax and examples demonstrating lasso, square-root lasso, and elastic net in

[LASSO] **lasso**	Lasso for prediction and model selection
[LASSO] **sqrtlasso**	Square-root lasso for prediction and model selection
[LASSO] **elasticnet**	Elastic net for prediction and model selection
[LASSO] **lasso examples**	Examples of lasso for prediction

If you are interested in lasso for inference, you can read more about the concepts, methods, and corresponding Stata commands in

[LASSO] **Lasso inference intro**	Introduction to inferential lasso models

If you want to see syntax for one of the lassos for inference commands, see

[LASSO] **dsregress**	Double-selection lasso linear regression
[LASSO] **dslogit**	Double-selection lasso logistic regression
[LASSO] **dspoisson**	Double-selection lasso Poisson regression
[LASSO] **poregress**	Partialing-out lasso linear regression
[LASSO] **pologit**	Partialing-out lasso logistic regression
[LASSO] **popoisson**	Partialing-out lasso Poisson regression
[LASSO] **poivregress**	Partialing-out lasso instrumental-variables regression
[LASSO] **xporegress**	Cross-fit partialing-out lasso linear regression
[LASSO] **xpologit**	Cross-fit partialing-out lasso logistic regression
[LASSO] **xpopoisson**	Cross-fit partialing-out lasso Poisson regression
[LASSO] **xpoivregress**	Cross-fit partialing-out lasso instrumental-variables regression
[LASSO] **lasso options**	Lasso options for inferential models

You might instead want to start with worked examples that demonstrate the lasso inference commands.

[LASSO] **Inference examples**	Examples and workflow for inference

If your inference involves estimating treatment effects, you can read about the lasso inference command that estimates the ATE, ATET, or POMs at

[CAUSAL] **telasso**   Treatment-effects estimation using lasso

Whether you are using lasso for prediction or for inference, you may want to learn more about the process of fitting lasso models and how you can make modifications to this process.

[LASSO] **lasso fitting**   The process (in a nutshell) of fitting lasso models

Acknowledgments

We thank Christian B. Hansen of the University of Chicago for his advice over the years and his amazingly fast responses to our numerous questions.

We thank Otilia Boldea of Tilburg University for her advice and for sharing her course materials on model-selection methods and big data.

We thank Damian Kozbur of the University of Zurich for his advice about how to select the lasso penalty parameter.

We thank Denis Chetverikov of the University of California at Los Angeles for conversations about the CV lasso.

We thank Victor Chernozhukov of the Massachusetts Institute of Technology and Alexandre Belloni of Duke University for their responses to our questions.

We thank Achim Ahrens of ETH Zürich and Immigration Policy Lab, Christian B. Hansen of the University of Chicago, and Mark E. Schaffer of Heriot-Watt University for their Stata commands `lasso2`, `cvlasso`, `rlasso`, `pdslasso`, `ivlasso`, and `lassologit`, which are discussed in Ahrens, Hansen, and Schaffer (2018, 2020). We also thank them for conversations about their approach to these methods.

We thank Wilbur Townsend of Harvard Business School for his Stata command `elasticregress`.

We thank Adrian Mander of the SPC (Statistical Process Control) for his Stata command `lars`.

We thank Jordi Sunyer and Mar Alvarez-Pedrerol of the ISGlobal Center for Research in Environmental Epidemiology and Raquel Garcia-Esteban of the RTI Health Solutions for sharing the data they used in Sunyer et al. (2017).

References

Ahrens, A., C. B. Hansen, and M. E. Schaffer. 2018. pdslasso: Stata module for post-selection and post-regularization OLS or IV estimation and inference. Statistical Software Components S458459, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s458459.html.

———. 2020. lassopack: Model selection and prediction with regularization regression in Stata. *Stata Journal* 20: 176–235.

Ahrens, A., C. B. Hansen, M. E. Schaffer, and T. Wiemann. 2024. ddml: Double/debiased machine learning in Stata. *Stata Journal* 24: 3–45.

Belloni, A., V. Chernozhukov, and C. B. Hansen. 2014. High-dimensional methods and inference on structural and treatment effects. *Journal of Economic Perspectives* 28: 29–50. https://doi.org/10.1257/jep.28.2.29.

Berk, R., L. D. Brown, A. Buja, K. Zhang, and L. Zhao. 2013. Valid post-selection inference. *Annals of Statistics* 41: 802–837. https://doi.org/10.1214/12-AOS1077.

Bühlmann, P., and S. van de Geer. 2011. *Statistics for High-Dimensional Data: Methods, Theory and Applications.* Berlin: Springer.

Cameron, A. C., and P. K. Trivedi. 2022. *Microeconometrics Using Stata.* 2nd ed. College Station, TX: Stata Press.

Dallakyan, A. 2022. graphiclasso: Graphical lasso for learning sparse inverse-covariance matrices. *Stata Journal* 22: 625–642.

Drukker, D. M., and D. Liu. 2019. An introduction to the lasso in Stata. *The Stata Blog: Not Elsewhere Classified.* https://blog.stata.com/2019/09/09/an-introduction-to-the-lasso-in-stata/.

Hastie, T. J., R. J. Tibshirani, and M. Wainwright. 2015. *Statistical Learning with Sparsity: The Lasso and Generalizations.* Boca Raton, FL: CRC Press. https://doi.org/10.1201/b18401.

Huang, W., Y. Wang, and L. Zhou. 2024. Identify latent group structures in panel data: The classifylasso command. *Stata Journal* 24: 46–71.

Javanmard, A., and A. Montanari. 2014. Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research* 15: 2869–2909.

Lee, J. D., D. L. Sun, Y. Sun, and J. E. Taylor. 2016. Exact post-selection inference, with application to the lasso. *Annals of Statistics* 44: 907–927. https://doi.org/10.1214/15-AOS1371.

Leeb, H., and B. M. Pötscher. 2005. Model selection and inference: Facts and fiction. *Econometric Theory* 21: 21–59. https://doi.org/10.1017/S0266466605050036.

———. 2006. Can one estimate the conditional distribution of post-model-selection estimators? *Annals of Statistics* 34: 2554–2591. https://doi.org/10.1214/009053606000000821.

———. 2008. Sparse estimators and the oracle property, or the return of Hodges' estimator. *Journal of Econometrics* 142: 201–211. https://doi.org/10.1016/j.jeconom.2007.05.017.

Schwarz, C. 2023. Estimating text regressions using txtreg_train. *Stata Journal* 23: 799–812.

Sunyer, J., E. Suades-González, R. García-Esteban, I. Rivas, J. Pujol, M. Alvarez-Pedrerol, J. Forns, X. Querol, and X. Basagaña. 2017. Traffic-related air pollution and attention in primary school children: Short-term association. *Epidemiology* 28: 181–189. https://doi.org/10.1097/EDE.0000000000000603.

Tibshirani, R. J. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, B ser., 58: 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

van de Geer, S., P. Bühlmann, Y. Ritov, and R. Dezeure. 2014. On asymptotically optimal confidence regions and tests for high-dimensional models. *Annals of Statistics* 42: 1166–1202. https://doi.org/10.1214/14-AOS1221.

Zhang, C.-H., and S. S. Zhang. 2014. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society*, B ser., 76: 217–242. https://doi.org/10.1111/rssb.12026.

Also see

[LASSO] **Lasso inference intro** — Introduction to inferential lasso models

[H2OML] **h2oml** — Introduction to commands for Stata integration with H2O machine learning