**xpopoisson** fits a lasso Poisson regression model and reports incidence-rate ratios along with standard errors, test statistics, and confidence intervals for specified covariates of interest. The cross-fit partialing-out method is used to estimate effects for these variables and to select from potential control variables to be included in the model.

**Quick start**

Report an incidence-rate ratio from a Poisson regression of \( y \) on \( d1 \), and include \( x1–x100 \) as potential control variables to be selected by lassos

```stata
xpopoisson y d1, controls(x1-x100)
```

As above, and estimate incidence-rate ratios for the levels of categorical \( d2 \)

```stata
xpopoisson y d1 i.d2, controls(x1-x100)
```

As above, but use 20 folds instead of 10 for cross-fitting

```stata
xpopoisson y d1 i.d2, controls(x1-x100) xfolds(20)
```

As above, but repeat the cross-fitting procedure 15 times, and average the results

```stata
xpopoisson y d1 i.d2, controls(x1-x100) xfolds(20) resample(15)
```

Use cross-validation (CV) instead of a plugin iterative formula to select the optimal \( \lambda^* \) in each lasso

```stata
xpopoisson y d1 i.d2, controls(x1-x100) selection(cv)
```

As above, and set a random-number seed for reproducibility

```stata
xpopoisson y d1 i.d2, controls(x1-x100) selection(cv) rseed(28)
```

Specify CV for the lasso for \( y \) only, with the stopping rule criterion turned off

```stata
xpopoisson y d1 i.d2, controls(x1-x100) lasso(y, selection(cv), stop(0))
```

As above, but apply the option to the lassos for \( y \), \( d1 \), and \( i.d2 \)

```stata
xpopoisson y d1 i.d2, controls(x1-x100) lasso(*, selection(cv), stop(0))
```

**Menu**

Statistics > Lasso > Lasso inferential models > Count outcomes > Cross-fit partialing-out Poisson model
xpopoisson depvar varsofinterest [if] [in],
    controls([alwaysvars] othervars) [options]

varsofinterest are variables for which coefficients and their standard errors are estimated.

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td></td>
</tr>
<tr>
<td>*controls([alwaysvars] othervars) alwaysvars and othervars make up the set of control variables; alwaysvars are always included; lassos choose whether to include or exclude othervars</td>
<td></td>
</tr>
<tr>
<td>selection(plugin)</td>
<td>use a plugin iterative formula to select an optimal value of the lasso penalty parameter $\lambda^*$ for each lasso; the default</td>
</tr>
<tr>
<td>selection(cv)</td>
<td>use CV to select an optimal value of the lasso penalty parameter $\lambda^*$ for each lasso</td>
</tr>
<tr>
<td>selection(adaptive)</td>
<td>use adaptive lasso to select an optimal value of the lasso penalty parameter $\lambda^*$ for each lasso</td>
</tr>
<tr>
<td>sqrtlasso</td>
<td>use square-root lassos for varsofinterest</td>
</tr>
<tr>
<td>xfolds(#)</td>
<td>use # folds for cross-fitting</td>
</tr>
<tr>
<td>resample[(#)</td>
<td>repeat sample splitting # times and average results</td>
</tr>
<tr>
<td>technique(dm11</td>
<td>use either double machine learning 1 (dm11) or double machine learning 2 (dm12) estimation technique; dm12 is the default</td>
</tr>
<tr>
<td>missingok</td>
<td>after fitting lassos, ignore missing values in any othervars not selected, and include these observations in the final model</td>
</tr>
<tr>
<td>offset(varname_o)</td>
<td>include varname_o in the lasso and model for depvar with its coefficient constrained to be 1</td>
</tr>
<tr>
<td>exposure(varname_e)</td>
<td>include ln(varname_e) in the lasso and model for depvar with its coefficient constrained to be 1</td>
</tr>
</tbody>
</table>

Reporting

| level(#) | set confidence level; default is level(95) |             |
| irr | report incidence-rate ratios; the default |             |
| coef | report estimated coefficients |             |
| display_options | control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling |             |

Optimization

| [no]log | display or suppress an iteration log |             |
| verbose | display a verbose iteration log |             |
| rseed(#) | set random-number seed |             |

Advanced

| lasso(varlist, lasso_options) | specify options for the lassos for variables in varlist; may be repeated |             |
| sqrtlasso(varlist, lasso_options) | specify options for square-root lassos for variables in varlist; may be repeated |             |
vce(robust)  robust VCE is the only VCE available
reestimate  refit the model after using lassoselect to select a different λ*
noheader  do not display the header on the coefficient table
coefflegend  display legend instead of statistics

*controls() is required.
varsofinterest, alwaysvars, and othervars may contain factor variables. Base levels of factor variables cannot be set for alwaysvars and othervars. See [U] 11.4.3 Factor variables.

vce(robust), reestimate, noheader, and coefflegend do not appear in the dialog box.
See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

controls([alwaysvars] othervars) specifies the set of control variables, which control for omitted variables. Control variables are also known as confounding variables. xpopoisson fits lassos for depvar and each of the varsofinterest. alwaysvars are variables that are always to be included in these lassos. alwaysvars are optional. othervars are variables that each lasso will choose to include or exclude. That is, each lasso will select a subset of othervars and other lassos will potentially select different subsets of othervars. controls() is required.

selection(plugin | cv | adaptive) specifies the selection method for choosing an optimal value of the lasso penalty parameter λ* for each lasso or square-root lasso estimation. Separate lassos are estimated for depvar and each variable in varsofinterest. Specifying selection() changes the selection method for all of these lassos. You can specify different selection methods for different lassos using the option lasso() or sqrtlasso(). When lasso() or sqrtlasso() is used to specify a different selection method for the lassos of some variables, they override the global setting made using selection() for the specified variables.

selection(plugin) is the default. It selects λ* based on a “plugin” iterative formula dependent on the data. See [LASSO] lasso options.

selection(cv) selects the λ* that gives the minimum of the CV function. See [LASSO] lasso options.

selection(adaptive) selects λ* using the adaptive lasso selection method. It cannot be specified when sqrtlasso is specified. See [LASSO] lasso options.

sqrtlasso specifies that square-root lassos be done rather than regular lassos for the varsofinterest. This option does not apply to depvar. Square-root lassos are linear models, and the lasso for depvar is always a Poisson lasso. The option lasso() can be used with sqrtlasso to specify that regular lasso be done for some variables, overriding the global sqrtlasso setting for these variables. See [LASSO] lasso options.

xfolds(#) specifies the number of folds for cross-fitting. The default is xfolds(10).

resample[(#)] specifies that sample splitting be repeated and results averaged. This reduces the effects of the randomness of sample splitting on the estimated coefficients. Not specifying resample or resample(#) is equivalent to specifying resample(1). In other words, by default no resampling is done. Specifying resample alone is equivalent to specifying resample(10). That is, sample splitting is repeated 10 times. For each sample split, lassos are computed. So when this option is not specified, lassos are repeated xfolds(#) times. But when resample(#) is specified, lassos
are repeated \( x\text{folds}(\#) \times \text{resample}(\#) \) times. Thus, while we recommend using \texttt{resample} to get final results, note that it can be an extremely time-consuming procedure.

\texttt{technique(dml1|dml2)} specifies which cross-fitting technique is used, either double machine learning 1 (dml1) or double machine learning 2 (dml2). For both techniques, the initial estimation steps are the same. The sample is split into \( K = x\text{folds}(\#) \) folds. Then, coefficients on the controls are estimated using only the observations not in the \( k \)th fold, for \( k = 1, 2, \ldots, K \). Moment conditions for the coefficients on the \texttt{varsofinterest} are formed using the observations in fold \( k \). The default technique, dml2, solves the moment conditions jointly across all the observations. The optional technique, dml1, solves the moment conditions in each fold \( k \) to produce \( K \) different estimates, which are then averaged to form a single vector of estimates. See \textit{Methods and formulas}.

\texttt{missingok} specifies that, after fitting lassos, the estimation sample be redefined based on only the nonmissing observations of variables in the final model. In all cases, any observation with missing values for \texttt{depvar}, \texttt{varsofinterest}, \texttt{alwaysvars}, and \texttt{othervars} is omitted from the estimation sample for the lassos. By default, the same sample is used for calculation of the coefficients of the \texttt{varsofinterest} and their standard errors.

When \texttt{missingok} is specified, the initial estimation sample is the same as the default, but the sample used for the calculation of the coefficients of the \texttt{varsofinterest} can be larger. Now observations with missing values for any \texttt{othervars} not selected will be added to the estimation sample (provided there are no missing values for any of the variables in the final model).

\texttt{missingok} may produce more efficient estimates when data are missing completely at random. It does, however, have the consequence that estimation samples can change when selected variables differ in models fit using different selection methods. That is, when \texttt{othervars} contain missing values, the estimation sample for a model fit using the default \texttt{selection(plugin)} will likely differ from the estimation sample for a model fit using, for example, \texttt{selection(cv)}.

\texttt{offset(varname_o)} specifies that \( \texttt{varname}_o \) be included in the lasso and model for \texttt{depvar} with its coefficient constrained to be 1.

\texttt{exposure(varname_e)} specifies that \( \ln(\texttt{varname}_e) \) be included in the lasso and model for \texttt{depvar} with its coefficient constrained to be 1.

\texttt{level(\#)}; see \textit{[R] Estimation options}.

\texttt{irr} reports estimated coefficients transformed to incidence-rate ratios, that is, \( e^{\alpha} \). Standard errors and confidence intervals are similarly transformed. \texttt{irr} is the default.

\texttt{coef} reports the estimated coefficients \( \alpha \) rather than the incidence-rate ratios, \( e^{\alpha} \). This option affects how results are displayed, not how they are estimated. \texttt{coef} may be specified at estimation or when replaying previously estimated results.

display_options: \texttt{noci}, \texttt{nopvalues}, \texttt{noomitted}, \texttt{vsquish}, \texttt{noemptycells}, \texttt{baselevels}, \texttt{allbaselevels}, \texttt{nofvlabel}, \texttt{fvwrap(\#)}, \texttt{fvweapon(style)}, \texttt{cformat(\%fmt)}, \texttt{pformat(\%fmt)}, \texttt{sformat(\%fmt)}, and \texttt{nolstretch}; see \textit{[R] Estimation options}.

\texttt{no}\log displays or suppresses a log showing the progress of the estimation. By default, one-line messages indicating when each lasso estimation begins are shown. Specify \texttt{verbose} to see a more detailed log.

\texttt{verbose} displays a verbose log showing the iterations of each lasso estimation. This option is useful when doing \texttt{selection(cv)} or \texttt{selection(adaptive)}. It allows you to monitor the progress
of the lasso estimations for these selection methods, which can be time consuming when there are many othervars specified in controls().

rseed(#) sets the random-number seed. This option can be used to reproduce results. rseed(#) is equivalent to typing set seed # prior to running xpopoisson. Random numbers are used to produce split samples for cross-fitting. So for all selection() options, if you want to reproduce your results, you must either use this option or use set seed. See [R] set seed.

### Advanced

**lasso(varlist, lasso_options)** lets you set different options for different lassos, or advanced options for all lassos. You specify a varlist followed by the options you want to apply to the lassos for these variables. varlist consists of one or more variables from depvar or varsofinterest. _all or * may be used to specify depvar and all varsofinterest. This option is repeatable as long as different variables are given in each specification. lasso_options are selection(...), grid(...), stop(#), tolerance(#), dtolerance(#), and cvtolerance(#). When lasso(varlist, selection(...)) is specified, it overrides any global selection() option for the variables in varlist. It also overrides the global sqrtlasso option for these variables. See [LASSO] lasso options.

**sqrtlasso(varlist, lasso_options)** works like the option lasso(), except square-root lassos for the variables in varlist are done rather than regular lassos. varlist consists of one or more variables from varsofinterest. Square-root lassos are linear models, and this option cannot be used with depvar. This option is repeatable as long as different variables are given in each specification. lasso_options are selection(...), grid(...), stop(#), tolerance(#), dtolerance(#), and cvtolerance(#). When sqrtlasso(varlist, selection(...)) is specified, it overrides any global selection() option for the variables in varlist. It also overrides the global sqrtlasso option for these variables. See [LASSO] lasso options.

The following options are available with xpopoisson but are not shown in the dialog box:

**vce(robust)** is the only variance–covariance estimator available for these models. The large-sample variance of the method of moments estimator for the partialed-out model has the sandwich structure, the same structure that would be termed a “robust” variance estimator. Only under special assumptions, like a correctly specified parametric model, can the robust form be simplified. No such assumptions are commonly used for partialed-out models, so only the robust VCE is available.

**reestimate** is an advanced option that refits the xpopoisson model based on changes made to the underlying lassos using lassoselect. After running xpopoisson, you can select a different \( \lambda^* \) for one or more of the lassos estimated by xpopoisson. After selecting \( \lambda^* \), you type xpopoisson, reestimate to refit the xpopoisson model based on the newly selected \( \lambda^* \)'s.

reestimate may be combined only with reporting options.

**noheader** prevents the coefficient table header from being displayed.

**coeflegend**; see [R] Estimation options.

### Remarks and examples

xpopoisson performs cross-fit partialing-out lasso Poisson regression. This command estimates incidence-rate ratios, standard errors, and confidence intervals and performs tests for variables of interest while using lassos to select from among potential control variables.
The Poisson regression model is

$$E[y|d, x] = \exp(d\alpha' + x\beta')$$

where $d$ are the variables for which we wish to make inferences and $x$ are the potential control variables from which the lassos select. \texttt{xpopoisson} estimates the $\alpha$ coefficients and reports the corresponding incidence-rate ratios, $e^{\alpha}$. However, cross-fit partialing-out does not provide estimates of the coefficients on the control variables ($\beta$) or their standard errors. No estimation results can be reported for $\beta$.

For an introduction to the cross-fit partialing-out lasso method for inference, as well as the double-selection and partialing-out methods, see \cite{LASSO} Lasso inference intro.

Examples that demonstrate how to use \texttt{xpopoisson} and the other lasso inference commands are presented in \cite{LASSO} Inference examples. In particular, we recommend reading 1 Overview for an introduction to the examples and to the \texttt{v1} command, which provides tools for working with the large lists of variables that are often included when using lasso methods. See 2 Fitting and interpreting inferential models for comparisons of the different methods of fitting inferential models that are available in Stata. Everything we say there about methods of selection is applicable to both linear and nonlinear models. See 4 Fitting inferential models to count outcomes. What is different? for examples and discussion specific to Poisson regression models. The primary difference from linear models involves interpreting the results.

If you are interested in digging deeper into the lassos that are used to select controls, see 5 Exploring inferential model lassos in \cite{LASSO} Inference examples.

### Stored results

\texttt{xpopoisson} stores the following in \texttt{e()}:

**Scalars**

- $e(N)$: number of observations
- $e(k\_varsofinterest)$: number of variables of interest
- $e(k\_controls)$: number of potential control variables
- $e(k\_controls\_sel)$: number of selected control variables
- $e(df)$: degrees of freedom for test of variables of interest
- $e(chi2)$: chi-squared
- $e(p)$: $p$-value for test of variables of interest
- $e(n\_xfolds)$: number of folds for cross-fitting
- $e(n\_resample)$: number of resamples
- $e(rank)$: rank of $e(V)$

**Macros**

- $e(cmd)$: \texttt{xpopoisson}
- $e(cmdline)$: command as typed
- $e(depvar)$: name of dependent variable
- $e(lasso\_depvars)$: names of dependent variables for all lassos
- $e(varsofinterest)$: variables of interest
- $e(controls)$: potential control variables
- $e(controls\_sel)$: selected control variables
- $e(model)$: \texttt{poisson}
- $e(title)$: title in estimation output
- $e(offset)$: linear offset variable
- $e(chi2type)$: Wald; type of chi-squared test
- $e(vce)$: \texttt{vcetype} specified in \texttt{vce()}
- $e(vcetype)$: title used to label Std. Err.
- $e(rngstate)$: random-number state used
- $e(properties)$: $b V$
Methods and formulas

xpopoisson implements cross-fit partialing-out lasso Poisson regression (XPOLPR) as described in Chernozhukov et al. (2018), where they derived two versions of cross-fitting that are known as double machine learning 1 (DML1) and double machine learning 2 (DML2). DML2 is the default method and corresponds with option technique(dml2). Specify option technique(dml1) to get DML1 instead.

Methods DML1 and DML2 have a similar structure. Each does the following:

1. Partitions the sample into $K$ folds.
2. Uses the post-lasso estimates computed using the observations not in a specific fold to fill in the moment conditions for the observations in that fold.

DML1 solves the moment conditions using the observations in each fold to produce $K$ different estimates and then averages these $K$ estimates to produce the final estimate for the coefficients of interest. DML2 uses all the observations to solve the moment conditions to produce a single final estimate for the coefficients of interest.

The $K$ folds are chosen once by default. Specify option resample(#) to have the $K$ folds randomly selected $#$ times. This resampling removes the dependence of the estimator on any specifically selected folds, at the cost of more computer time. See Methods and formulas in [LASSO] xporegress for details about resampling.

The regression model is

$$E[y|d, x] = G(d\alpha' + \beta_0 + x\beta')$$

where $G(a) = \exp(a)$, $d$ contains the $J$ covariates of interest, and $x$ contains the $p$ controls. The number of covariates in $d$ must be small and fixed. The number of controls in $x$ can be large and, in theory, can grow with the sample size; however, the number of nonzero elements in $\beta$ must not be too large, which is to say that the model must be sparse.

XPOLPR algorithm

1. Randomly partition the sample into $K$ subsamples called folds.
2. Define $I_k$ to be the observations in fold $k$, and define $IC_k$ to be the sample observations not in fold $k$.
3. For each $k = 1, \ldots, K$, fill in the observations of $i \in I_k$ for the $J$ moment conditions that identify $\alpha$. These moment conditions use out-of-sample estimates of the high-dimensional components estimated using the observations $i \in IC_k$.
   a. Using the observations $i \in IC_k$, perform a Poisson lasso of $y$ on $d$ and $x$ to select controls $\tilde{x}_{k,y}$.

This Poisson lasso can choose the lasso penalty parameter ($\lambda^*_k$) using the plugin estimator, adaptive lasso, or CV. The plugin value is the default.
b. Using the observations \( i \in IC_k \), fit a Poisson regression of \( y \) on \( d \) and \( \tilde{x}_{k,y} \), let \( \tilde{\alpha}_k \) be the estimated coefficients on \( d \), and let \( \tilde{\delta}_k \) be the estimated coefficients on \( \tilde{x}_{k,y} \).

c. For the observations \( i \in I_k \), fill in the prediction for the high-dimensional component using the out-of-sample estimate \( \tilde{\delta}_k \).

\[
\tilde{s}_i = \tilde{x}_{k,y,i} \tilde{\delta}_k'
\]

d. Using the observations \( i \in IC_k \), for \( j = 1, \ldots, J \), perform a linear lasso of \( d_j \) on \( x \) using observation-level weights

\[
w_i = G'(d_i \tilde{\alpha}_k' + \tilde{s}_i)
\]

where \( G'() \) is the derivative of \( G() \), and denote the selected controls by \( \tilde{x}_{k,j} \). Each of these lassos can choose the lasso penalty parameter (\( \lambda_j^* \)) using one of the plugin estimators for a linear lasso, adaptive lasso, or CV. The heteroskedastic plugin estimator for the linear lasso is the default.

e. Using the observations \( i \in IC_k \), for \( j = 1, \ldots, J \), fit a linear regression of \( d_j \) on \( \tilde{x}_{k,j} \), and denote the coefficient estimates by \( \tilde{\gamma}_{k,j} \).

f. For each observation \( i \in I_k \), and for \( j = 1, \ldots, J \), fill in the instrument

\[
z_{j,i} = d_{j,i} - \tilde{x}_{k,j,i} \tilde{\gamma}_{k,j}
\]

g. For each observation \( i \in I_k \), collect the instruments into a vector \( z_i = (z_{1,i}, z_{2,i}, \ldots, z_{J,i}) \).

4. Compute the point estimates.

For DML2, compute \( \hat{\alpha} \) by solving the following sample-moment equations.

\[
\frac{1}{n} \sum_{i=1}^{n} \{y_i - G(d_i \alpha' + \tilde{s}_i)\} z_i' = 0
\]

For DML1, \( \hat{\alpha} \) is given by

\[
\hat{\alpha} = \frac{1}{K} \sum_{k=1}^{K} \hat{\alpha}_k
\]

where \( \hat{\alpha}_k \) is computed by solving the sample-moment equations

\[
\frac{1}{n_k} \sum_{i \in I_k} \{y_i - G(d_i \alpha_k' + \tilde{s}_i)\} z_i' = 0
\]

and \( n_k \) is the number of observations in \( I_k \).
5. The VCE is estimated by

\[
\widehat{\text{Var}}(\hat{\alpha}) = \frac{1}{n} \widehat{J}_0^{-1} \hat{\Psi} \left( \widehat{J}_0^{-1} \right)'
\]

where

\[
\hat{\Psi} = \frac{1}{K} \sum_{k=1}^{K} \hat{\Psi}_k
\]

\[
\hat{\Psi}_k = \frac{1}{n_k} \sum_{i \in I_k} \hat{\psi}_i \hat{\psi}_i'
\]

\[
\hat{\psi}_i = \{ y_i - G(d_i \hat{\alpha}' + \hat{s}_i) \} z_i'
\]

\[
\hat{J}_0 = \frac{1}{K} \sum_{k=1}^{K} \left( \frac{1}{n_k} \sum_{i \in I_k} \hat{\psi}_a i \right)
\]

and

\[
\hat{\psi}_a = \frac{\partial \hat{\psi}_i}{\partial \hat{\alpha}}
\]

See Methods and formulas in [LASSO lasso] for details on how the lassos in steps 3a and 3d choose their penalty parameters (\(\lambda^*\)).

Reference


Also see

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

[LASSO] dpoisson — Double-selection lasso Poisson regression

[LASSO] popoisson — Partialing-out lasso Poisson regression

[R] poisson — Poisson regression

[U] 20 Estimation and postestimation commands