

zip — Zero-inflated Poisson regression

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

`zip` estimates a zero-inflated Poisson (ZIP) regression of *depvar* on *indepvars*, where *depvar* is a nonnegative count variable.

Quick start

Zero-inflated Poisson model of *y* on *x1* and *x2* with inflation modeled using *x3*

```
zip y x1 x2, inflate(x3)
```

And conduct Vuong test of ZIP model against a standard Poisson model

```
zip y x1 x2, inflate(x3) vuong
```

Use a probit model instead of a logit model to predict excess zeros

```
zip y x1 x2, inflate(x3) probit
```

Menu

Statistics > Count outcomes > Zero-inflated Poisson regression

Syntax

```
zip depvar [indepvars] [if] [in] [weight] ,
      inflate(varlist [ , offset(varname) ] | _cons) [options]
```

<i>options</i>	Description
Model	
* <u>inflate</u> ()	equation that determines whether the count is zero
<u>noconstant</u>	suppress constant term
<u>exposure</u> (<i>varname</i> _{<i>e</i>})	include $\ln(\text{varname}_e)$ in model with coefficient constrained to 1
<u>offset</u> (<i>varname</i> _{<i>o</i>})	include <i>varname</i> _{<i>o</i>} in model with coefficient constrained to 1
<u>constraints</u> (<i>constraints</i>)	apply specified linear constraints
<u>collinear</u>	keep collinear variables
<u>probit</u>	use probit model to characterize excess zeros; default is logit
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>oim</u> , <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <u>opg</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is <u>level</u> (95)
<u>irr</u>	report incidence-rate ratios
<u>vuong</u>	perform Vuong test
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>coeflegend</u>	display legend instead of statistics

* inflate(*varlist* [, offset(*varname*)] | _cons) is required.

indepvars and *varlist* may contain factor variables; see [U] 11.4.3 Factor variables.

bootstrap, by, fp, jackknife, rolling, statsby, and svy are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

vce(), vuong, and weights are not allowed with the svy prefix; see [SVY] svy.

fweights, iweights, and pweights are allowed; see [U] 11.1.6 weight.

coeflegend does not appear in the dialog box.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

Options

Model

inflate(*varlist* [, offset(*varname*)] | _cons) specifies the equation that determines whether the observed count is zero. Conceptually, omitting inflate() would be equivalent to fitting the model with poisson; see [R] poisson.

`inflate(varlist [, offset(varname)])` specifies the variables in the equation. You may optionally include an offset for this *varlist*.

`inflate(_cons)` specifies that the equation determining whether the count is zero contains only an intercept. To run a zero-inflated model of *depvar* with only an intercept in both equations, type `zip depvar, inflate(_cons)`.

`noconstant`, `exposure(varnamee)`, `offset(varnameo)`, `constraints(constraints)`, `collinear`; see [R] [estimation options](#).

`probit` requests that a probit, instead of logit, model be used to characterize the excess zeros in the data.

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] [vce_option](#).

Reporting

`level(#)`; see [R] [estimation options](#).

`irr` reports estimated coefficients transformed to incidence-rate ratios, that is, e^b rather than b . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated or stored. `irr` may be specified at estimation or when replaying previously estimated results.

`vuong` specifies that the [Vuong \(1989\)](#) test of ZIP versus Poisson be reported. This test statistic has a standard normal distribution with large positive values favoring the ZIP model and large negative values favoring the Poisson model.

`nocnsreport`; see [R] [estimation options](#).

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [estimation options](#).

Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrto`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

Setting the optimization type to `technique(bhhh)` resets the default *vcetype* to `vce(opg)`.

The following option is available with `zip` but is not shown in the dialog box:

`coeflegend`; see [R] [estimation options](#).

Remarks and examples

[stata.com](http://www.stata.com)

See [Long \(1997, 242–247\)](#) and [Greene \(2012, 821–826\)](#) for a discussion of zero-modified count models. For information about the test developed by [Vuong \(1989\)](#), see [Greene \(2012, 823–824\)](#) and [Long \(1997\)](#). [Greene \(1994\)](#) applied the test to ZIP and ZINB models, as described in [Greene \(2012, 824\)](#).

Poisson regression fits models of the number of occurrences (counts) of an event. You could use `poisson` for this (see [R] [poisson](#)), but in some count-data models, you might want to account for the prevalence of zero counts in the data.

For instance, you might count how many fish each visitor to a park catches. Many visitors may catch zero, because they do not fish (as opposed to being unsuccessful). You may be able to model whether a person fishes depending on several covariates related to fishing activity and model how many fish a person catches depending on several covariates having to do with the success of catching fish (type of lure/bait, time of day, temperature, season, etc.). This is the type of data for which the `zip` command is useful.

The zero-inflated (or zero-altered) Poisson model allows overdispersion through the splitting process that models the outcomes as zero or nonzero.

▷ Example 1

We have data on the number of fish caught by visitors to a national park. Some of the visitors do not fish, but we do not have the data on whether a person fished; we merely have data on how many fish were caught together with several covariates. Because our data have a preponderance of zeros (142 of 250), we use the `zip` command to model the outcome.

```
. use http://www.stata-press.com/data/r14/fish
. zip count persons livebait, inf(child camper) vuong
Fitting constant-only model:
Iteration 0:  log likelihood = -1347.807
              (output omitted)
Iteration 4:  log likelihood = -1103.9425
Fitting full model:
Iteration 0:  log likelihood = -1103.9425
              (output omitted)
Iteration 5:  log likelihood = -850.70142
Zero-inflated Poisson regression
Inflation model = logit
Log likelihood = -850.7014
```

	Number of obs	=	250
	Nonzero obs	=	108
	Zero obs	=	142
	LR chi2(2)	=	506.48
	Prob > chi2	=	0.0000

	count	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
	count						
	persons	.8068853	.0453288	17.80	0.000	.7180424	.8957281
	livebait	1.757289	.2446082	7.18	0.000	1.277866	2.236713
	_cons	-2.178472	.2860289	-7.62	0.000	-2.739078	-1.617865
	inflate						
	child	1.602571	.2797719	5.73	0.000	1.054228	2.150913
	camper	-1.015698	.365259	-2.78	0.005	-1.731593	-.2998038
	_cons	-.4922872	.3114562	-1.58	0.114	-1.10273	.1181558

```
Vuong test of zip vs. standard Poisson: z = 3.95 Pr>z = 0.0000
```

In general, Vuong test statistics that are significantly positive favor the zero-inflated models, while those that are significantly negative favor the nonzero-inflated models. Thus, in the above model, the zero inflation is significant.

Stored results

zip stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_zero)</code>	number of zero observations
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model
<code>e(ll_c)</code>	log likelihood, comparison model
<code>e(df_c)</code>	degrees of freedom for comparison test
<code>e(N_clust)</code>	number of clusters
<code>e(chi2)</code>	χ^2
<code>e(p)</code>	significance of model test
<code>e(vuong)</code>	Vuong test statistic
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	zip
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(inflate)</code>	logit or probit
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset1)</code>	offset
<code>e(offset2)</code>	offset for <code>inflate()</code>
<code>e(chi2type)</code>	Wald or LR; type of model χ^2 test
<code>e(vce)</code>	<i>vcetype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log (up to 20 iterations)
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

Methods and formulas

Several models in the literature are (correctly) described as zero inflated. The `zip` command maximizes the log-likelihood $\ln L$, defined by

$$\begin{aligned}\xi_j^\beta &= \mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j^\beta \\ \xi_j^\gamma &= \mathbf{z}_j\boldsymbol{\gamma} + \text{offset}_j^\gamma \\ \ln L &= \sum_{j \in S} w_j \ln [F(\xi_j^\gamma) + \{1 - F(\xi_j^\gamma)\} \exp(-\lambda_j)] + \\ &\quad \sum_{j \notin S} w_j \left[\ln\{1 - F(\xi_j^\gamma)\} - \lambda_j + \xi_j^\beta y_j - \ln(y_j!) \right]\end{aligned}$$

where w_j are the weights, F is the inverse of the logit link (or the inverse of the probit link if `probit` was specified), and S is the set of observations for which the outcome $y_j = 0$.

This command supports the Huber/White/sandwich estimator of the variance and its clustered version using `vce(robust)` and `vce(cluster clustvar)`, respectively. See [P] [_robust](#), particularly *Maximum likelihood estimators* and *Methods and formulas*.

`zip` also supports estimation with survey data. For details on VCEs with survey data, see [SVY] [variance estimation](#).

References

- Desmarais, B. A., and J. J. Harden. 2013. [Testing for zero inflation in count models: Bias correction for the Vuong test](#). *Stata Journal* 13: 810–835.
- Greene, W. H. 1994. Accounting for excess zeros and sample selection in Poisson and negative binomial regression models. Working paper EC-94-10, Department of Economics, Stern School of Business, New York University. <http://ideas.repec.org/p/ste/nystbu/94-10.html>.
- . 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.
- Lambert, D. 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34: 1–14.
- Long, J. S. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage.
- Long, J. S., and J. Freese. 2001. [Predicted probabilities for count models](#). *Stata Journal* 1: 51–57.
- . 2014. *Regression Models for Categorical Dependent Variables Using Stata*. 3rd ed. College Station, TX: Stata Press.
- Mullahy, J. 1986. Specification and testing of some modified count data models. *Journal of Econometrics* 33: 341–365.
- Vuong, Q. H. 1989. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica* 57: 307–333.

Also see

- [R] **zip postestimation** — Postestimation tools for zip
- [R] **zinb** — Zero-inflated negative binomial regression
- [R] **nbreg** — Negative binomial regression
- [R] **poisson** — Poisson regression
- [R] **tnbreg** — Truncated negative binomial regression
- [R] **tpoisson** — Truncated Poisson regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models
- [U] **20 Estimation and postestimation commands**