zinb — Zero-inflated negative binomial regression

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

Description

zinb estimates a zero-inflated negative binomial (ZINB) regression of *depvar* on *indepvars*, where *depvar* is a nonnegative count variable.

Quick start

Zero-inflated negative binomial model of y on x1 and x2 with inflation modeled using x3

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

And conduct Vuong test of ZINB model against standard negative binomial model and likelihood-ratio test against ZIP model

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

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

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

Menu

Statistics > Count outcomes > Zero-inflated negative binomial regression

Syntax

zinb *depvar* [*indepvars*] [*if*] [*in*] [*weight*] ,

inflate(*varlist*[, offset(*varname*)]|_cons) [*options*]

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

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 zinb *depvar*, inflate(_cons).

noconstant, exposure(*varname*$_e$), offset(*varname*$_o$), 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^{β_i} rather than β_i. 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 ZINB versus negative binomial be reported. This test statistic has a standard normal distribution with large positive values favoring the ZINB model and large negative values favoring the negative binomial model.

zip requests that a likelihood-ratio test comparing the ZINB model with the zero-inflated Poisson model be included in the output.

nocnsreport; see [R] **estimation options**.

display_options: noci, 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(*#*), nonrtolerance, 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 zinb but is not shown in the dialog box:

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

Remarks and examples

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 zero-inflated Poisson and negative binomial models, and there is a description of that work in Greene (2012).

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

For instance, you could 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 `zinb` command is useful.

The zero-inflated (or zero-altered) negative binomial 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 have data merely on how many fish were caught, together with several covariates. Because our data have a preponderance of zeros (142 of 250), we use the `zinb` command to model the outcome.

```
. use http://www.stata-press.com/data/r14/fish
. zinb count persons livebait, inf(child camper) vuong
Fitting constant-only model:
Iteration 0:   log likelihood = -519.33992
  (output omitted )
Iteration 8:   log likelihood = -442.66299
Fitting full model:
Iteration 0:   log likelihood = -442.66299  (not concave)
  (output omitted )
Iteration 8:   log likelihood = -401.54776
Zero-inflated negative binomial regression      Number of obs     =        250
                                                Nonzero obs       =        108
                                                Zero obs          =        142
Inflation model = logit                         LR chi2(2)        =      82.23
Log likelihood  = -401.5478                     Prob > chi2       =     0.0000

------------------------------------------------------------------------------
       count |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
count        |
     persons |   .9742984   .1034938     9.41   0.000     .7714543    1.177142
    livebait |   1.557523   .4124424     3.78   0.000     .7491503    2.365895
       _cons |  -2.730064    .476953    -5.72   0.000    -3.664874   -1.795253
-------------+----------------------------------------------------------------
inflate      |
       child |   3.185999   .7468551     4.27   0.000      1.72219    4.649808
      camper |  -2.020951    .872054    -2.32   0.020    -3.730146   -.3117567
       _cons |  -2.695385   .8929071    -3.02   0.003     -4.44545   -.9453189
-------------+----------------------------------------------------------------
    /lnalpha |   .5110429   .1816816     2.81   0.005     .1549535    .8671323
-------------+----------------------------------------------------------------
       alpha |   1.667029   .3028685                      1.167604    2.380076
------------------------------------------------------------------------------
Vuong test of zinb vs. standard negative binomial: z =     5.59  Pr>z = 0.0000
```

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

Stored results

`zinb` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_zero)`	number of zero observations
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_aux)`	number of auxiliary parameters
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(df_c)`	degrees of freedom for comparison test
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(p)`	significance of model test
`e(chi2_cp)`	χ^2 for test of $\alpha = 0$
`e(vuong)`	Vuong test statistic
`e(rank)`	rank of `e(V)`
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`zinb`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(inflate)`	`logit` or `probit`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset1)`	offset
`e(offset2)`	offset for `inflate()`
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_cpt)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_cp)`
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. Err.
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

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

Functions

`e(sample)`	marks estimation sample

Methods and formulas

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

$$
\begin{aligned}
m &= 1/\alpha \\
p_j &= 1/(1+\alpha\mu_j) \\
\xi_j^{\beta} &= \mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j^{\beta} \\
\xi_j^{\gamma} &= \mathbf{z}_j\boldsymbol{\gamma} + \text{offset}_j^{\gamma} \\
\mu_j &= \exp(\xi_j^{\beta}) \\
\ln L &= \sum_{j\in S} w_j \ln\left[F(\xi_j^{\gamma}) + \{1-F(\xi_j^{\gamma})\}p_j^m\right] \\
&\qquad + \sum_{j\notin S} w_j \Big[\ln\{1-F(\xi_j^{\gamma})\} + \ln\Gamma(m+y_j) - \ln\Gamma(y_j+1) \\
&\qquad\qquad - \ln\Gamma(m) + m\ln p_j + y_j\ln(1-p_j)\Big]
\end{aligned}
$$

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*.

`zinb` 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.

Harris, T., J. M. Hilbe, and J. W. Hardin. 2014. Modeling count data with generalized distributions. *Stata Journal* 14: 562–579.

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] **zinb postestimation** — Postestimation tools for zinb

[R] **zip** — Zero-inflated Poisson 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] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

[U] **20 Estimation and postestimation commands**