**Syntax**

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

**options**

- `* 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 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.
Menu

Statistics > Count outcomes > Zero-inflated negative binomial regression

Description

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

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, \( e^{\beta_i} \) rather than \( \beta_i \). 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`: `nomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(\%fmt)`, `pformat(\%fmt)`, `sformat(\%fmt)`, and `nolstretch`; see [R] estimation options.
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 default dialog box: coeflegend; see [R] estimation options.

Remarks and examples


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.
```stata
use http://www.stata-press.com/data/r13/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

|                | 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 non–zero-inflated models. Thus, in the above model, the zero inflation is significant.
```
Stored results

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

\textbf{Scalars}

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

\textbf{Macros}

- \texttt{e(cmd)}: \texttt{zinb}  
- \texttt{e(cmdline)}: command as typed  
- \texttt{e(depvar)}: name of dependent variable  
- \texttt{e(inflate)}: logit or probit  
- \texttt{e(wtype)}: weight type  
- \texttt{e(wexp)}: weight expression  
- \texttt{e(title)}: title in estimation output  
- \texttt{e(clustvar)}: name of estimation variable  
- \texttt{e(offset1)}: offset  
- \texttt{e(offset2)}: offset for \texttt{inflate()}  
- \texttt{e(chi2type)}: Wald or LR; type of model $\chi^2$ test  
- \texttt{e(chi2_cpt)}: Wald or LR; type of model $\chi^2$ test corresponding to \texttt{e(chi2_cp)}  
- \texttt{e(vcetype)}: vcetype specified in \texttt{vce()}  
- \texttt{e(title)}: title used to label Std. Err.  
- \texttt{e(opt)}: type of optimization  
- \texttt{e(ml_method)}: type of ml method  
- \texttt{e(user)}: name of likelihood-evaluator program  
- \texttt{e(properties)}: b V  
- \texttt{e(predict)}: program used to implement \texttt{predict}  
- \texttt{e(asbalanced)}: factor variables \texttt{fvset} as \texttt{asbalanced}  
- \texttt{e(asobserved)}: factor variables \texttt{fvset} as \texttt{asobserved}

\textbf{Matrices}

- \texttt{e(b)}: coefficient vector  
- \texttt{e(Cns)}: constraints matrix  
- \texttt{e(gradient)}: iteration log (up to 20 iterations)  
- \texttt{e(V)}: variance–covariance matrix of the estimators  
- \texttt{e(V_modelbased)}: model-based variance

\textbf{Functions}

- \texttt{e(sample)}: marks estimation sample
Methods and formulas

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

$$
m = 1/\alpha$$
$$p_j = 1/(1 + \alpha \mu_j)$$
$$\xi_j^\beta = x_j \beta + \text{offset}_j^\beta$$
$$\xi_j^\gamma = z_j \gamma + \text{offset}_j^\gamma$$
$$\mu_j = \exp(\xi_j^\beta)$$
$$\ln L = \sum_{j \in S} w_j \ln \left[ F(\xi_j^\gamma) + \{1 - F(\xi_j^\gamma)\} p_j^m \right]$$
$$+ \sum_{j \not\in S} w_j \left[ \ln\{1 - F(\xi_j^\gamma)\} + \ln\Gamma(m + y_j) - \ln\Gamma(y_j + 1) - \ln\Gamma(m) + m \ln p_j + y_j \ln(1 - p_j) \right]$$

where $w_j$ are the weights, $F$ is the inverse of the logit link (or the inverse of the probit link if \texttt{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 \texttt{vce(robust)} and \texttt{vce(cluster clustvar)}, respectively. See \cite{P_robust}, particularly \textit{Maximum likelihood estimators} and \textit{Methods and formulas}.

\texttt{zinb} also supports estimation with survey data. For details on VCEs with survey data, see \cite[SVY]{} \textbf{variance estimation}.

References


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