### Syntax

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

**options**

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>*inflate()</code></td>
<td>equation that determines whether the count is zero</td>
</tr>
<tr>
<td><code>noconstant</code></td>
<td>suppress constant term</td>
</tr>
<tr>
<td><code>exposure(varname_o)</code></td>
<td>include ( \ln(varname_o) ) in model with coefficient constrained to 1</td>
</tr>
<tr>
<td><code>offset(varname_o)</code></td>
<td>include ( varname_o ) in model with coefficient constrained to 1</td>
</tr>
<tr>
<td><code>constraints(constraints)</code></td>
<td>apply specified linear constraints</td>
</tr>
<tr>
<td><code>collinear</code></td>
<td>keep collinear variables</td>
</tr>
<tr>
<td><code>probit</code></td>
<td>use probit model to characterize excess zeros; default is logit</td>
</tr>
</tbody>
</table>

| SE/Robust           | vcetype may be oim, robust, cluster clustvar, opg, bootstrap, or jackknife |

<table>
<thead>
<tr>
<th>Reporting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>level(#)</code></td>
<td>set confidence level; default is level(95)</td>
</tr>
<tr>
<td><code>irr</code></td>
<td>report incidence-rate ratios</td>
</tr>
<tr>
<td><code>vuong</code></td>
<td>perform Vuong test</td>
</tr>
<tr>
<td><code>nocnsreport</code></td>
<td>do not display constraints</td>
</tr>
<tr>
<td><code>display_options</code></td>
<td>control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>maximize_options</code></td>
<td>control the maximization process; seldom used</td>
</tr>
<tr>
<td><code>coeflegend</code></td>
<td>display legend instead of statistics</td>
</tr>
</tbody>
</table>

* `*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.
Description

zip estimates a zero-inflated Poisson (ZIP) 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 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(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, \( e^b \) 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. \( \text{irr} \) 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: nomet, vsquish, noemptycells, baselevels, allbaselevels, nolabel, fvwrap(#), fvwrapon(style), cformat(%fmt), pformat(%fmt), sformat(%fmt), and nolstretch; see [R] estimation options.
**Remarks and examples**


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/r13/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

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

`zip` 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 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(ll_c)` log likelihood, comparison model
- `e(df_c)` degrees of freedom for comparison test
- `e(N_clust)` number of clusters
- `e(chi2)` \( \chi^2 \)
- `e(p)` significance of model test
- `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)` `zip`
- `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 \( \chi^2 \) test
- `e(vce)` `vcetype` specified in `vce()`
- `e(vcetype)` title used to label Std. Err.
- `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 `zip` command maximizes the log-likelihood \( \ln L \), defined by

\[
\begin{align*}
\xi_j^\beta &= x_j \beta + \text{offset}_j^\beta \\
\xi_j^\gamma &= z_j \gamma + \text{offset}_j^\gamma \\
\ln L &= \sum_{j \in S} w_j \ln \left[ F(\xi_j^\gamma) + \{1 - F(\xi_j^\gamma)\} \exp(-\lambda_j) \right] + \sum_{j \notin S} w_j \left[ \ln \{1 - F(\xi_j^\gamma)\} - \lambda_j + \xi_j^\beta y_j - \ln(y_j!) \right]
\end{align*}
\]

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 \textit{Maximum likelihood estimators and Methods and formulas}.

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

### References


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