

**zioprobit** — Zero-inflated ordered probit regression

|                             |                                      |                                |                                      |
|-----------------------------|--------------------------------------|--------------------------------|--------------------------------------|
| <a href="#">Description</a> | <a href="#">Quick start</a>          | <a href="#">Menu</a>           | <a href="#">Syntax</a>               |
| <a href="#">Options</a>     | <a href="#">Remarks and examples</a> | <a href="#">Stored results</a> | <a href="#">Methods and formulas</a> |
| <a href="#">References</a>  | <a href="#">Also see</a>             |                                |                                      |

## Description

`zioprobit` fits a model for a discrete ordered outcome with a high fraction of zeros, called zero inflation. This model is known as a zero-inflated ordered probit (ZIOP) model. In the context of ZIOP models, zero is an actual 0 value or the lowest outcome category. The ZIOP model accounts for the zero inflation by assuming that the zero-valued outcomes come from both a probit model and an ordered probit model, allowing potentially different sets of covariates for each model.

## Quick start

Zero-inflated ordered probit model of `y` on `x1` and categorical variable `a` with excess zeros modeled using `x2`

```
zioprobit y x1 i.a, inflate(x2)
```

Add offset `x3` to the model

```
zioprobit y x1 i.a, inflate(x2) offset(x3)
```

Model excess zeros using only a constant

```
zioprobit y x1 i.a, inflate(_cons)
```

Model excess zeros with `x2`, and offset `x5` while suppressing the constant term

```
zioprobit y x1 i.a, inflate(x2, offset(x5) noconstant)
```

Account for complex sampling design using `svyset` data

```
svy: zioprobit y x1 i.a, inflate(x2)
```

## Menu

Statistics > Ordinal outcomes > Zero-inflated ordered probit regression

## Syntax

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

| <i>options</i>   | Description  |
|--|--|
| Model  |  |
| * <u>inflate</u> ()  | equation that determines excess zero values  |
| <u>offset</u> ( <i>varname</i> )   | include <i>varname</i> in model with coefficient constrained to 1  |
| <u>constraints</u> ( <i>constraints</i> )  | apply specified linear constraints   |
| <u>collinear</u>   | keep collinear variables   |
| 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>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   |
| * <u>inflate</u> ( <i>varlist</i> [, <u>noconstant</u> <u>offset</u> ( <i>varname</i> )]   <u>_cons</u> ) is required.   |  |
| <i>indepvars</i> and <i>varlist</i> may contain factor variables; see [U] 11.4.3 <b>Factor variables</b> .   |  |
| <u>bayes</u> , <u>bootstrap</u> , <u>by</u> , <u>fp</u> , <u>jackknife</u> , <u>rolling</u> , <u>statsby</u> , and <u>svy</u> are allowed; see [U] 11.1.10 <b>Prefix commands</b> .<br>For more details, see [BAYES] <b>bayes: zioprobit</b> . |  |
| Weights are not allowed with the <u>bootstrap</u> prefix; see [R] <b>bootstrap</b> .   |  |
| <u>vce</u> () and weights are not allowed with the <u>svy</u> prefix; see [SVY] <b>svy</b> .   |  |
| <u>fweights</u> , <u>iweights</u> , and <u>pweights</u> are allowed; see [U] 11.1.6 <b>weight</b> .  |  |
| <u>coeflegend</u> does not appear in the dialog box.   |  |
| See [U] 20 <b>Estimation and postestimation commands</b> for more capabilities of estimation commands.   |  |

## Options

---

Model

inflate(*varlist* [, noconstant offset(*varname*)] | \_cons) specifies the equation that determines the excess zero values; this option is required. Conceptually, omitting inflate() would be equivalent to fitting the model with oprobit; see [R] **oprobit**.

inflate(*varlist* [, noconstant offset(*varname*)] ) specifies the variables in the equation that determines the excess zeros. To suppress the constant in this equation, specify the noconstant suboption. You may optionally include an offset for this *varlist*.

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

`offset(varname)`, `constraints(constraints)`, `collinear`; see [R] [estimation options](#).

#### 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](#).

`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(#)`, `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 `zioprobit` but is not shown in the dialog box:

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

## Remarks and examples

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

ZIOP models are used when the outcome of interest is an ordered response variable and the data exhibit a high fraction of observations in the lowest category or what we will refer to from now on as “zero” but without quotes. Like the ordered probit model, the actual values taken by the ordered response variable are irrelevant. While the outcome is typically coded as 0, 1, 2, . . . , *H*, `zioprobit` interprets the lowest value present in the dataset as 0 to be consistent with the original derivation of the model and subsequent applications.

Like all zero-inflated models, the ZIOP model is an alternative when the data exhibit a higher fraction of zero-valued outcomes than is likely compatible with an ordered probit model. This concentration of zeros is referred to as zero inflation. Inflation is assumed to occur in the lowest value to ensure that shifting the levels of the ordered response variable by a constant will not affect the estimated parameters in the model. This is common in ordered probit models; see [R] [oprobit](#).

Without loss of generality, we consider an ordered response variable with levels 0, 1, 2, . . . , *H*. Traditional ordered probit models treat all observations with zero-valued outcomes as a homogeneous group. By contrast, ZIOP models assume that zeros could occur in the data as members of two latent (unobservable) groups. Individuals in the “always-zero” group have outcome 0 as the only possible value. This first group is often called the nonparticipation group. The second group, in addition to 0,

may also assume any of the other values, 1, 2, . . . ,  $H$ . This group is often called the participation group. Some disciplines, such as public health and medicine, refer to the process that determines the zeros rather than the groups. In this case, there is an incidence or occurrence process that determines whether an observation belongs to the always-zero group and a severity or intensity process that determines the level in the second group. The result of having two groups or processes is an inflation in the proportion of zero-valued observations in the data.

The ZIOP model has been used in studies of international and domestic conflicts (Bagozzi et al. 2015), sports participation (Downward, Lera-Lopez, and Rasciute 2011), and the adoption of new building technologies (Ganguly, Koebel, and Cantrell 2010), to name a few. See Kelley and Anderson (2008) for a discussion of zero-inflated ordinal models in the context of health.

The classic application of the ZIOP model is the study of tobacco use by Harris and Zhao (2007). Like the zero-inflated Poisson models in the count-data literature (Lambert 1992), Harris and Zhao derived the ZIOP model using a two-stage decision process. An individual must decide whether to participate in an activity (for example, smoking or drug consumption) and, conditional on participating, must decide on the level of participation, which also includes zero participation. The first decision is a binary choice and is modeled using a probit model, while the second is an ordered choice and is modeled using an ordered probit model. In other terms, to account for the excess of zeros, Harris and Zhao allowed for zero observations to occur in two ways: as a realization of the probit model (nonparticipants) and as a realization of the ordered probit model when the binary random variable in the probit model is 1 (participant with zero activity). See *Methods and formulas* for more details. For a Bayesian derivation of the ZIOP model, see Gurme and Gagne (2009).

You may want to check whether a standard ordered probit model is adequate to fit the data. You can do this, for instance, using information criteria; see [example 2](#) in `[R] zip`.

### ► Example 1: Zero-inflated ordered probit model

We have fictional data on cigarette consumption per day for 15,000 subjects between ages 14 and 84. The outcome of interest, `tobacco`, is an ordered discrete response with four levels coded as 0 for “no cigarettes”, 1 for “up to 8 cigarettes/day”, 2 for “8 to 12 cigarettes/day”, and 3 for “more than 12 cigarettes/day”. The exact number of daily consumed cigarettes is unknown.

About 63% of the respondents identified themselves as current nonsmokers. We suspect that these self-identified current nonsmokers belong to one of two groups. Individuals in the first group are genuine nonsmokers (always-zero group) who have never smoked and will never smoke. Individuals in the second group are smokers with no smoking activity who could be the corner solution of a standard consumer demand problem and who may smoke, say, if the price of tobacco falls or their income increases. It is likely that these two types of zeros are driven by different patterns of consumer behavior and a ZIOP model is a good candidate in this case.

We model tobacco consumption levels for subjects who choose to smoke as a function of years of education (`education`), annual income in tens of thousands of dollars (`income`), age in tens of years (`age`), and whether the respondent is a female (`female`). In addition to `education`, `income`, `age`, and `female`, the decision to smoke is modeled as a function of whether either of the respondent’s parents smoked (`parent`) and whether the respondent’s religion discourages smoking (`religion`). We list all the covariates in the required `inflate()` option.

```
. use http://www.stata-press.com/data/r15/tobacco
(Fictional tobacco consumption data)
. zioprobit tobacco education income i.female age,
> inflate(education income i.parent age i.female i.religion)
```

```
Iteration 0: log likelihood = -14820.211 (not concave)
Iteration 1: log likelihood = -12819.475 (not concave)
Iteration 2: log likelihood = -12078.843 (not concave)
Iteration 3: log likelihood = -10926.037
Iteration 4: log likelihood = -9549.5112
Iteration 5: log likelihood = -8662.3141
Iteration 6: log likelihood = -7749.9803
Iteration 7: log likelihood = -7647.1348
Iteration 8: log likelihood = -7640.5027
Iteration 9: log likelihood = -7640.4738
Iteration 10: log likelihood = -7640.4738
```

```
Zero-inflated ordered probit regression
Number of obs = 15,000
Wald chi2(4) = 2574.27
Prob > chi2 = 0.0000
Log likelihood = -7640.4738
```

| tobacco      | Coef.     | Std. Err. | z      | P> z  | [95% Conf. Interval] |           |
|--------------|-----------|-----------|--------|-------|----------------------|-----------|
| tobacco      |           |           |        |       |                      |           |
| education    | .5112664  | .0102407  | 49.92  | 0.000 | .491195              | .5313378  |
| income       | .712975   | .0144803  | 49.24  | 0.000 | .6845942             | .7413559  |
| female       |           |           |        |       |                      |           |
| female       | -.3975341 | .0416675  | -9.54  | 0.000 | -.4792009            | -.3158674 |
| age          | -.7709896 | .0182554  | -42.23 | 0.000 | -.8067695            | -.7352097 |
| inflate      |           |           |        |       |                      |           |
| education    | -.0966613 | .0026422  | -36.58 | 0.000 | -.1018398            | -.0914827 |
| income       | -.1157545 | .0043787  | -26.44 | 0.000 | -.1243365            | -.1071725 |
| parent       |           |           |        |       |                      |           |
| smoking      | .7655798  | .0307553  | 24.89  | 0.000 | .7053006             | .825859   |
| age          | .1873904  | .0088643  | 21.14  | 0.000 | .1700168             | .204764   |
| female       |           |           |        |       |                      |           |
| female       | -.2639665 | .0307184  | -8.59  | 0.000 | -.3241735            | -.2037595 |
| religion     |           |           |        |       |                      |           |
| discourage.. | -.3223335 | .0496827  | -6.49  | 0.000 | -.4197098            | -.2249572 |
| _cons        | 1.27051   | .0584794  | 21.73  | 0.000 | 1.155892             | 1.385127  |
| /cut1        | 2.959808  | .0753035  |        |       | 2.812216             | 3.1074    |
| /cut2        | 8.111228  | .1648965  |        |       | 7.788037             | 8.43442   |
| /cut3        | 11.20791  | .2247711  |        |       | 10.76736             | 11.64845  |

In the output table, the first set of coefficients, labeled `tobacco`, corresponds to the participation (smoking) levels. These coefficients are interpreted in the same way as coefficients from an ordered probit model. The second set of coefficients, labeled `inflate`, corresponds to the equation for the participation decision. These are interpreted in the same way as coefficients from a binary probit model. ZIOP models do not require the variables to be the same in the participation level and decision equations. However, the same variables can appear in both. If the same variables are included, it is not uncommon for the coefficients to have opposite signs. For example, `income` and `education` in the model above have positive signs in the level equation and negative signs in the decision equation.

The estimated coefficients are not particularly informative, and as with all discrete choice models, marginal effects are better to interpret. We use `margins` to estimate the average marginal effect of

having a smoking parent on the probability of being a genuine nonsmoker. Specifying `pnp` within the `predict()` option means that we are requesting the predicted probability of nonparticipation, which in our example is equivalent to being a genuine nonsmoker.

```
. margins, predict(pnp) dydx(parent)
Average marginal effects          Number of obs   =       15,000
Model VCE      : OIM
Expression    : Pr(nonparticipation), predict(pnp)
dy/dx w.r.t.  : 1.parent
```

|                   | Delta-method |           |        |       |                      |
|-------------------|--------------|-----------|--------|-------|----------------------|
|                   | dy/dx        | Std. Err. | z      | P> z  | [95% Conf. Interval] |
| parent<br>smoking | -.1797895    | .0071967  | -24.98 | 0.000 | -.1938948 -.1656843  |

Note: dy/dx for factor levels is the discrete change from the base level.

On average, individuals whose parents are smokers are about 18% less likely to be nonsmokers themselves than individuals whose parents did not use tobacco. See [R] [zioprobit postestimation](#) for more information and examples.

4

## Stored results

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

### Scalars

|                            |   |
|----------------------------|---|
| <code>e(N)</code>          | number of observations                          |
| <code>e(N_zero)</code>     | number of zeros or lowest-category observations |
| <code>e(k_cat)</code>      | number of categories                            |
| <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_aux)</code>      | number of auxiliary parameters                  |
| <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(N_clust)</code>    | number of clusters                              |
| <code>e(chi2)</code>       | $\chi^2$  |
| <code>e(p)</code>          | <i>p</i> -value for model test                  |
| <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>            | <code>zioprobit</code>   |
| <code>e(cmdline)</code>        | command as typed   |
| <code>e(depvar)</code>         | name of dependent variable   |
| <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 $\chi^2$ test                                  |
| <code>e(vce)</code>            | <code>vcetype</code> 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(marginsok)</code>      | predictions allowed by <code>margins</code>                              |
| <code>e(marginsnotok)</code>   | predictions disallowed by <code>margins</code>                           |
| <code>e(marginsdefault)</code> | default <code>predict()</code> specification for <code>margins</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(cat)</code>          | category values                              |
| <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

Consider a discrete ordered response variable  $Y$  with levels coded as  $0, 1, 2, \dots, H$ . For notational simplicity, we assume that the zeros are inflated, but the following derivation may be adapted to accommodate inflation in the lowest outcome category. Harris and Zhao (2007) derived the ZIOP model in two steps. First, the group membership (participants versus nonparticipants) can be modeled using a probit model. Let  $s_j = 1$  if the  $j$ th individual belongs to the participation group or let  $s_j = 0$  otherwise. With the probit model, the probability of participation is given by

$$\Pr(s_j = 1 | \mathbf{z}_j) = \Phi(\mathbf{z}_j \boldsymbol{\gamma}) \quad (1)$$

$\mathbf{z}_j$  is a vector of covariates that determines group membership,  $\boldsymbol{\gamma}$  is a vector of coefficients that have to be estimated, and  $\Phi(\cdot)$  is the standard normal distribution function. Next, conditioning on  $s_j = 1$ , participation levels  $\tilde{y}_j$  are modeled using an ordered probit model; these levels may also include 0. The corresponding probabilities are given by

$$\Pr(\tilde{y}_j = h | \mathbf{x}_j, s_j = 1) = \Phi(\kappa_h - \mathbf{x}_j \boldsymbol{\beta}) - \Phi(\kappa_{h-1} - \mathbf{x}_j \boldsymbol{\beta}) \quad h = 0, 1, \dots, H \quad (2)$$

where  $\kappa_{-1} = -\infty$ ,  $\kappa_H = +\infty$ , and  $\mathbf{x}_j$  is a vector of covariates that could be different from  $\mathbf{z}_j$ .  $\kappa_h$  are boundary parameters that need to be estimated in addition to the coefficients vector  $\boldsymbol{\beta}$ .

The intercept  $\beta_0$  is set equal to 0 in (2) for identification. Note that  $s_j$  and  $\tilde{y}_j$  are both unobservable in terms of the zeros. The observed response variable is  $y_j = s_j \tilde{y}_j$ . Thus, the zero outcome occurs when  $s_j = 0$  (the individual is not a participant) or occurs when  $s_j = 1$  and  $\tilde{y}_j = 0$  (the individual is a participant with zero activity). To observe a positive  $y_j$ , it is a joint requirement that  $s_j = 1$  and  $\tilde{y}_j > 0$ .

The distribution of  $Y$  is given by

$$\begin{aligned} \Pr(Y) &= \begin{cases} \Pr(y_j = 0 | \mathbf{z}_j, \mathbf{x}_j) \\ \Pr(y_j = h | \mathbf{z}_j, \mathbf{x}_j) \end{cases} \quad h = 1, 2, \dots, H \\ &= \begin{cases} \Pr(s_j = 0 | \mathbf{z}_j) + \Pr(s_j = 1 | \mathbf{z}_j) \Pr(\tilde{y}_j = 0 | \mathbf{x}_j, s_j = 1) \\ \Pr(s_j = 1 | \mathbf{z}_j) \Pr(\tilde{y}_j = h | \mathbf{x}_j, s_j = 1) \end{cases} \quad h = 1, 2, \dots, H \end{aligned} \tag{3}$$

The probability of zero outcome has been inflated because it is the sum of the probability of zero activity from the ordered probit model and the probability of nonparticipation from the probit model.

Substituting (1) and (2) in (3), we get

$$\begin{aligned} \Pr(Y) &= \begin{cases} \Pr(y_j = 0 | \mathbf{z}_j, \mathbf{x}_j) \\ \Pr(y_j = h | \mathbf{z}_j, \mathbf{x}_j) \\ \Pr(y_j = H | \mathbf{z}_j, \mathbf{x}_j) \end{cases} \quad h = 1, 2, \dots, H - 1 \\ &= \begin{cases} \{1 - \Phi(\mathbf{z}_j \boldsymbol{\gamma})\} + \Phi(\mathbf{z}_j \boldsymbol{\gamma}) \Phi(\kappa_0 - \mathbf{x}_j \boldsymbol{\beta}) \\ \Phi(\mathbf{z}_j \boldsymbol{\gamma}) \{ \Phi(\kappa_h - \mathbf{x}_j \boldsymbol{\beta}) - \Phi(\kappa_{h-1} - \mathbf{x}_j \boldsymbol{\beta}) \} \\ \Phi(\mathbf{z}_j \boldsymbol{\gamma}) \{ 1 - \Phi(\kappa_{H-1} - \mathbf{x}_j \boldsymbol{\beta}) \} \end{cases} \quad h = 1, 2, \dots, H - 1 \end{aligned}$$

If the respective `offset()` option is specified,  $\mathbf{x}_j \boldsymbol{\beta}$  and  $\mathbf{z}_j \boldsymbol{\gamma}$  are replaced with  $\mathbf{x}_j \boldsymbol{\beta} + \text{offset}_j^\beta$  and  $\mathbf{z}_j \boldsymbol{\gamma} + \text{offset}_j^\gamma$ , respectively.

The log-likelihood function is

$$\ln L = \sum_{j=1}^N w_j \sum_{h=0}^H I(y_j = h) \ln \{ \Pr(y_j = h | \mathbf{z}_j, \mathbf{x}_j) \}$$

where  $w_j$  is an optional weight for the  $j$ th observation and

$$I(y_j = h) = \begin{cases} 1 & \text{if } y_j = h \\ 0 & \text{otherwise} \end{cases}$$

The choice between the ZIOP model and the ordered probit model cannot be made using a likelihood-ratio test because the two hypotheses are not nested in the usual sense of parameter restrictions. The restriction  $\gamma = 0$  does not eliminate the inflation effect; it makes the group membership probabilities both equal to 0.5 [see (1)]. What is needed to remove the inflation effect is  $\mathbf{z}_j \boldsymbol{\gamma} \rightarrow \infty$ , which cannot be imposed. Because ZIOP and ordered probit models are not nested, you can compare the fits of the two models using information criteria.

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

`zioprobit` also supports estimation with survey data. For details on VCEs with survey data, see [SVY] **variance estimation**.



## References

- Bagozzi, B. E., D. W. Hill, Jr, W. H. Moore, and B. Mukherjee. 2015. Modeling two types of peace: The zero-inflated ordered probit (ZiOP) model in conflict research. *Journal of Conflict Resolution* 59: 728–752.
- Downward, P., F. Lera-Lopez, and S. Rasciute. 2011. The zero-inflated ordered probit approach to modelling sports participation. *Economic Modelling* 28: 2469–2477.
- Ganguly, I., C. T. Koebel, and R. A. Cantrell. 2010. A categorical modeling approach to analyzing new product adoption and usage in the context of the building-materials industry. *Technological Forecasting and Social Change* 77: 662–677.
- Greene, W. H., and D. A. Hensher. 2010. *Modeling Ordered Choices: A Primer*. New York: Cambridge University.
- Gurme, S., and G. A. Gagne. 2009. Bayesian Approach to Zero-Inflated Ordered Probit Models, with an Application. Presented at the 2009 Seminar on Bayesian Inference in Econometrics and Statistics. [https://www.maxwell.syr.edu/uploadedFiles/econ/seminars/ZIOP\\_GurmuDagne.pdf](https://www.maxwell.syr.edu/uploadedFiles/econ/seminars/ZIOP_GurmuDagne.pdf).
- Harris, M. N., and X. Zhao. 2007. A zero-inflated ordered probit model, with an application to modelling tobacco consumption. *Journal of Econometrics* 141: 1073–1099.
- Kelley, M. E., and S. J. Anderson. 2008. Zero inflation in ordinal data: Incorporating susceptibility to response through the use of a mixture model. *Statistics in Medicine* 27: 3674–3688.
- Lambert, D. 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34: 1–14.
- Long, J. S., and J. Freese. 2014. *Regression Models for Categorical Dependent Variables Using Stata*. 3rd ed. College Station, TX: Stata Press.

## Also see

- [R] [zioprobit postestimation](#) — Postestimation tools for zioprobit
- [R] [oprobit](#) — Ordered probit regression
- [R] [probit](#) — Probit regression
- [BAYES] [bayes: zioprobit](#) — Bayesian zero-inflated ordered probit regression
- [SVY] [svy estimation](#) — Estimation commands for survey data
- [U] [20 Estimation and postestimation commands](#)