

**eoprobit** — Extended 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

`eoprobit` fits an ordered probit regression model that accommodates any combination of endogenous covariates, nonrandom treatment assignment, and endogenous sample selection. Continuous, binary, and ordinal endogenous covariates are allowed. Treatment assignment may be endogenous or exogenous. A probit or tobit model may be used to account for endogenous sample selection.

`xteoprobit` fits a random-effects ordered probit regression model that accommodates endogenous covariates, treatment, and sample selection in the same way as `eoprobit` and also accounts for correlation of observations within panels or within groups.

## Quick start

- Ordered probit regression of  $y$  on  $x$  with continuous endogenous covariate  $y_2$  modeled by  $x$  and  $z$
- ```
eoprobit y x, endogenous(y2 = x z)
```
- Same as above, but adding continuous endogenous covariate  $y_3$  modeled by  $x$  and  $z_2$
- ```
eoprobit y x, endogenous(y2 = x z) endogenous(y3 = x z2)
```
- Ordered probit regression of  $y$  on  $x$  with binary endogenous covariate  $d$  modeled by  $x$  and  $z$
- ```
eoprobit y x, endogenous(d = x z, probit)
```
- Ordered probit regression of  $y$  on  $x$  with endogenous treatment recorded in `trtvar` and modeled by  $x$  and  $z$
- ```
eoprobit y x, entreat(trtvar = x z)
```
- Ordered probit regression of  $y$  on  $x$  with exogenous treatment recorded in `trtvar`
- ```
eoprobit y x, extreat(trtvar)
```
- Random-effects ordered probit regression of  $y$  on  $x$  using `xtset` data
- ```
xteoprobit y x
```
- Ordered probit regression of  $y$  on  $x$  with endogenous sample-selection indicator `selvar` modeled by  $x$  and  $z$
- ```
eoprobit y x, select(selvar = x z)
```
- Same as above, but adding endogenous covariate  $y_2$  modeled by  $x$  and  $z_2$
- ```
eoprobit y x, select(selvar = x z) endogenous(y2 = x z2)
```
- Same as above, but adding endogenous treatment recorded in `trtvar` and modeled by  $x$  and  $z_3$
- ```
eoprobit y x, select(selvar = x z) endogenous(y2 = x z2) ///
    entreat(trtvar = x z3)
```

Same as above, but with random effects and without endogenous treatment

```
xteoprobit y x, select(selvar = x z) endogenous(y2 = x z2)
```

## Menu

### eoprobit

Statistics > Endogenous covariates > Models adding selection and treatment > Ordered probit regression

### xteoprobit

Statistics > Longitudinal/panel data > Endogenous covariates > Models adding selection and treatment > Ordered probit regression (RE)

## Syntax

*Basic ordered probit regression with endogenous covariates*

```
eoprobit depvar [indepvars], endogenous(depvarsen = varlisten) [options]
```

*Basic ordered probit regression with endogenous treatment assignment*

```
eoprobit depvar [indepvars], entreat(depvartr [= varlisttr]) [options]
```

*Basic ordered probit regression with exogenous treatment assignment*

```
eoprobit depvar [indepvars], extreat(tvar) [options]
```

*Basic ordered probit regression with sample selection*

```
eoprobit depvar [indepvars], select(depvars = varlists) [options]
```

*Basic ordered probit regression with tobit sample selection*

```
eoprobit depvar [indepvars], tobitselect(depvars = varlists) [options]
```

*Basic ordered probit regression with random effects*

```
xteoprobit depvar [indepvars] [, options]
```

*Ordered probit regression combining endogenous covariates, treatment, and selection*

```
eoprobit depvar [indepvars] [if] [in] [weight] [, extensions options]
```

*Ordered probit regression combining random effects, endogenous covariates, treatment, and selection*

```
xteoprobit depvar [indepvars] [if] [in] [, extensions options]
```

| <i>extensions</i>                            | Description                                                                                                                                      |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Model                                        |                                                                                                                                                  |
| <u>endogenous</u> ( <i>enspec</i> )          | model for endogenous covariates; may be repeated                                                                                                 |
| <u>entreat</u> ( <i>entrspec</i> )           | model for endogenous treatment assignment                                                                                                        |
| <u>extreat</u> ( <i>extrspec</i> )           | exogenous treatment                                                                                                                              |
| <u>select</u> ( <i>selspec</i> )             | probit model for selection                                                                                                                       |
| <u>tobitselect</u> ( <i>tselspec</i> )       | tobit model for selection                                                                                                                        |
| <hr/>                                        |                                                                                                                                                  |
| <i>options</i>                               | Description                                                                                                                                      |
| Model                                        |                                                                                                                                                  |
| <u>offset</u> ( <i>varname<sub>o</sub></i> ) | include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1                                                                    |
| <u>constraints</u> ( <i>numlist</i> )        | apply specified linear constraints                                                                                                               |
| 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 |
| Integration                                  |                                                                                                                                                  |
| <u>intpoints</u> (#)                         | set the number of integration (quadrature) points for integration over four or more dimensions; default is <u>intpoints</u> (128)                |
| <u>triintpoints</u> (#)                      | set the number of integration (quadrature) points for integration over three dimensions; default is <u>triintpoints</u> (10)                     |
| <u>reintpoints</u> (#)                       | set the number of integration (quadrature) points for random-effects integration; default is <u>reintpoints</u> (7)                              |
| <u>reintmethod</u> ( <i>intmethod</i> )      | integration method for random effects; <i>intmethod</i> may be <u>mvaghermite</u> (the default) or <u>ghermite</u>                               |
| Maximization                                 |                                                                                                                                                  |
| <u>maximize_options</u>                      | control the maximization process; seldom used                                                                                                    |
| <u>collinear</u>                             | keep collinear variables                                                                                                                         |
| <u>coeflegend</u>                            | display legend instead of statistics                                                                                                             |

*enspec* is  $depvars_{en} = varlist_{en} [ , enopts ]$

where  $depvars_{en}$  is a list of endogenous covariates. Each variable in  $depvars_{en}$  specifies an endogenous covariate model using the common  $varlist_{en}$  and options.

*entrspec* is  $depvar_{tr} [= varlist_{tr} ] [ , entropts ]$

where  $depvar_{tr}$  is a variable indicating treatment assignment.  $varlist_{tr}$  is a list of covariates predicting treatment assignment.

#### 4 eoprobit — Extended ordered probit regression

---

*extrspec* is *tvar* [ , *extropts* ]

where *tvar* is a variable indicating treatment assignment.

*selspec* is *depvar<sub>s</sub>* = *varlist<sub>s</sub>* [ , *selopts* ]

where *depvar<sub>s</sub>* is a variable indicating selection status. *depvar<sub>s</sub>* must be coded as 0, indicating that the observation was not selected, or 1, indicating that the observation was selected. *varlist<sub>s</sub>* is a list of covariates predicting selection.

*tselspec* is *depvar<sub>s</sub>* = *varlist<sub>s</sub>* [ , *tseopts* ]

where *depvar<sub>s</sub>* is a continuous variable. *varlist<sub>s</sub>* is a list of covariates predicting *depvar<sub>s</sub>*. The censoring status of *depvar<sub>s</sub>* indicates selection, where a censored *depvar<sub>s</sub>* indicates that the observation was not selected and a noncensored *depvar<sub>s</sub>* indicates that the observation was selected.

---

| <i>enopts</i> | Description |
|---------------|-------------|
|---------------|-------------|

---

Model

|                      |                                                                                               |
|----------------------|-----------------------------------------------------------------------------------------------|
| <u>probit</u>        | treat endogenous covariate as binary                                                          |
| <u>oprobit</u>       | treat endogenous covariate as ordinal                                                         |
| <u>pocorrelation</u> | estimate different correlations for each level of a binary or an ordinal endogenous covariate |
| <u>nomain</u>        | do not add endogenous covariate to main equation                                              |
| <u>nore</u>          | do not include random effects in model for endogenous covariate                               |
| <u>noconstant</u>    | suppress constant term                                                                        |

---

nore is available only with xteoprobit.

---

| <i>entropts</i> | Description |
|-----------------|-------------|
|-----------------|-------------|

---

Model

|                                              |                                                                               |
|----------------------------------------------|-------------------------------------------------------------------------------|
| <u>pocorrelation</u>                         | estimate different correlations for each potential outcome                    |
| <u>nomain</u>                                | do not add treatment indicator to main equation                               |
| <u>nocutsinteract</u>                        | do not interact treatment with cutpoints                                      |
| <u>nointeract</u>                            | do not interact treatment with covariates in main equation                    |
| <u>nore</u>                                  | do not include random effects in model for endogenous treatment               |
| <u>noconstant</u>                            | suppress constant term                                                        |
| <u>offset</u> ( <i>varname<sub>o</sub></i> ) | include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1 |

---

nore is available only with xteoprobit.

---

| <i>extropts</i> | Description |
|-----------------|-------------|
|-----------------|-------------|

---

Model

|                       |                                                            |
|-----------------------|------------------------------------------------------------|
| <u>pocorrelation</u>  | estimate different correlations for each potential outcome |
| <u>nomain</u>         | do not add treatment indicator to main equation            |
| <u>nocutsinteract</u> | do not interact treatment with cutpoints                   |
| <u>nointeract</u>     | do not interact treatment with covariates in main equation |

---

| <i>selopts</i>                           | Description                                                                   |
|------------------------------------------|-------------------------------------------------------------------------------|
| Model                                    |                                                                               |
| <code>nore</code>                        | do not include random effects in selection model                              |
| <code>noconstant</code>                  | suppress constant term                                                        |
| <code>offset(varname<sub>o</sub>)</code> | include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1 |

`nore` is available only with `xteoprobit`.

| <i>tselopts</i>                          | Description                                                                   |
|------------------------------------------|-------------------------------------------------------------------------------|
| Model                                    |                                                                               |
| <code>*ll(varname   #)</code>            | left-censoring variable or limit                                              |
| <code>*ul(varname   #)</code>            | right-censoring variable or limit                                             |
| <code>main</code>                        | add censored selection variable to main equation                              |
| <code>nore</code>                        | do not include random effects in tobit selection model                        |
| <code>noconstant</code>                  | suppress constant term                                                        |
| <code>offset(varname<sub>o</sub>)</code> | include <i>varname<sub>o</sub></i> in model with coefficient constrained to 1 |

\* You must specify either `ll()` or `ul()`.

`nore` is available only with `xteoprobit`.

*indepvars*, *varlist<sub>en</sub>*, *varlist<sub>tr</sub>*, and *varlist<sub>s</sub>* may contain factor variables; see [U] 11.4.3 Factor variables.

*devar*, *indepvars*, *depvars<sub>en</sub>*, *varlist<sub>en</sub>*, *devar<sub>tr</sub>*, *varlist<sub>tr</sub>*, *tvar*, *devar<sub>s</sub>*, and *varlist<sub>s</sub>* may contain time-series operators; see [U] 11.4.4 Time-series varlists.

`bootstrap`, `by`, `collect`, `jackknife`, and `statsby` are allowed with `eoprobit` and `xteoprobit`. `rolling` and `svy` are allowed with `eoprobit`. See [U] 11.1.10 Prefix commands.

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

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

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

`reintpoints()` and `reintmethod()` are available only with `xteoprobit`.

`collinear` and `coeflegend` do not appear in the dialog box.

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

## Options

### Model

`endogenous(enspec)`, `entreat(entspec)`, `extreat(extrspec)`, `select(selspec)`, `tobitselect(tselspec)`, `re`; see [ERM] ERM options.

`offset(varnameo)`, `constraints(numlist)`; see [R] Estimation options.

### SE/Robust

`vce(vcetype)`; see [ERM] ERM options.

### Reporting

`level(#)`, `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.

## Integration

`intpoints(#)`, `triintpoints(#)`, `reintpoints(#)`, `reintmethod(intmethod)`; see [\[ERM\] ERM 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](#).

The default technique for `eoprobit` is `technique(nr)`. The default technique for `xteoprobit` is `technique(bhhh 10 nr 2)`.

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

The following options are available with `eoprobit` and `xteoprobit` but are not shown in the dialog box:

`collinear`, `coeflegend`; see [\[R\] Estimation options](#).

## Remarks and examples

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

`eoprobit` and `xteoprobit` fit models that we refer to as “extended ordered probit regression models”, meaning that they accommodate endogenous covariates, nonrandom treatment assignment, endogenous sample selection, and panel data or other grouped data.

`eoprobit` fits models for cross-sectional data (one-level models). `eoprobit` can account for endogenous covariates, treatment, and sample selection, whether these complications arise individually or in combination.

`xteoprobit` fits random-effects models (two-level models) for panel data or grouped data. `xteoprobit` accounts for endogenous covariates, treatment, and sample selection in the same way as `eoprobit` and also accounts for within-panel or within-group correlation among observations.

In this entry, you will find information on the syntax for the `eoprobit` and `xteoprobit` commands. You can see [Methods and formulas](#) for a full description of the models that can be fit with `eoprobit` and `xteoprobit` and details about how those models are fit.

More information on extended ordered probit models is found in the separate introductions and example entries. We recommend reading those entries to learn how to use `eoprobit` and `xteoprobit`. Below, we provide a guide to help you locate the ones that will be helpful to you.

For an introduction to `eoprobit` and `xteoprobit` and the other extended regression commands for continuous, interval, and binary outcomes, see [\[ERM\] Intro 1](#)–[\[ERM\] Intro 9](#).

[\[ERM\] Intro 1](#) introduces the ERM commands, the problems they address, and their syntax.

[\[ERM\] Intro 2](#) provides background on the four types of models—linear regression, interval regression, probit regression, and ordered probit regression—that can be fit using ERM commands.

[\[ERM\] Intro 3](#) considers the problem of endogenous covariates and how to solve it using ERM commands.

[\[ERM\] Intro 4](#) gives an overview of endogenous sample selection and using ERM commands to account for it when fitting a linear, interval, probit, or ordered probit model.

[\[ERM\] Intro 5](#) covers nonrandom treatment assignment and how to account for it using `eoprobit` or any of the other ERM commands.

[ERM] **Intro 6** covers random-effects models for panel data and other grouped data. It discusses `xteoprobit` and the other ERM commands for panel data.

[ERM] **Intro 7** discusses interpretation of results. You can interpret coefficients from `eoprobit` and `xteoprobit` in the usual way, but this introduction goes beyond the interpretation of coefficients. We demonstrate how to find answers to interesting questions by using `margins`. If your model includes an endogenous covariate or an endogenous treatment, the use of `margins` differs from its use after other estimation commands, so we strongly recommend reading this intro if you are fitting these types of models.

[ERM] **Intro 8** will be particularly helpful if you are familiar with `heckoprobit` or `xtoprobit` and other commands that address endogenous covariates, sample selection, nonrandom treatment assignment, or random effects. This introduction is a Rosetta stone that maps the syntax of those commands to the syntax of `eoprobit` and `xteoprobit`.

[ERM] **Intro 9** walks you through an example that gives insight into the concepts of endogenous covariates, treatment assignment, and sample selection while fitting models with `eregress` that address these complications. Although the example uses `eregress`, the discussion applies equally to `eoprobit`. This intro also demonstrates how to interpret results by using `margins` and `estat teffects`.

Additional examples are presented in [ERM] **Example 1a**–[ERM] **Example 9**. For examples using `eoprobit`, see

|                         |                                                                          |
|-------------------------|--------------------------------------------------------------------------|
| [ERM] <b>Example 6a</b> | Ordered probit regression with endogenous treatment                      |
| [ERM] <b>Example 6b</b> | Ordered probit regression with endogenous treatment and sample selection |

For an example using `xteoprobit`, see

|                        |                                                                        |
|------------------------|------------------------------------------------------------------------|
| [ERM] <b>Example 9</b> | Ordered probit regression with endogenous treatment and random effects |
|------------------------|------------------------------------------------------------------------|

See *Examples* in [ERM] **Intro** for an overview of all the examples. All examples may be interesting because they handle complications in the same way.

`eoprobit` and `xteoprobit` fit many models discussed in the literature. For instance, `eoprobit` can be used to fit models like the ordered probit model with endogenous sample selection discussed in De Luca and Perotti (2011) and the ordered probit models with continuous or binary endogenous covariates discussed in Wooldridge (2010, sec. 16.3.3). Roodman (2011) investigated ordered probit models with endogenous covariates and endogenous sample selection and demonstrated how multiple observational data complications could be addressed with a triangular model structure. He and Tamás Bartus showed how random effects could be used in the triangular model structure in Bartus and Roodman (2014). Roodman’s work has been used to model processes like the effect of living with a child on the happiness of the elderly (Chyi and Mao 2012) and the effect of parental migration on child education (Botezat and Pfeiffer 2014).

## Stored results

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

Scalars

|                               |                                                                |
|-------------------------------|----------------------------------------------------------------|
| <code>e(N)</code>             | number of observations                                         |
| <code>e(N_selected)</code>    | number of selected observations                                |
| <code>e(N_nonselected)</code> | number of nonselected observations                             |
| <code>e(k)</code>             | number of parameters                                           |
| <code>e(k_cat#)</code>        | number of categories for the <code>#th depvar</code> , ordinal |
| <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(k_aux)</code>      | number of auxiliary parameters                       |
| <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(n_quad)</code>     | number of integration points for multivariate normal |
| <code>e(n_quad3)</code>    | number of integration points for trivariate normal   |
| <code>e(rank)</code>       | rank of $e(V)$                                       |
| <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>            | <b>eoprobit</b>                                                                                     |
| <code>e(cmdline)</code>        | command as typed                                                                                    |
| <code>e(depvar)</code>         | names of dependent variables                                                                        |
| <code>e(tsel_ll)</code>        | left-censoring limit for tobit selection                                                            |
| <code>e(tsel_ul)</code>        | right-censoring limit for tobit selection                                                           |
| <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(offset#)</code>        | offset for the <i>#</i> th <i>depvar</i> , where <i>#</i> is determined by equation order in output |
| <code>e(chi2type)</code>       | Wald; type of model $\chi^2$ test                                                                   |
| <code>e(vce)</code>            | <i>vce</i> type 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>     | b V                                                                                                 |
| <code>e(estat_cmd)</code>      | program used to implement <code>estat</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(cat#)</code>         | categories for the <i>#</i> th <i>depvar</i> , ordinal |
| <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 |
|------------------------|-------------------------|

In addition to the above, the following is stored in `r()`:

## Matrices

|                       |                                                                                                                            |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------|
| <code>r(table)</code> | matrix containing the coefficients with their standard errors, test statistics, <i>p</i> -values, and confidence intervals |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------|

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any *r*-class command is run after the estimation command.



xteoprobit stores the following in `e()`:

#### Scalars

|                               |                                                          |
|-------------------------------|----------------------------------------------------------|
| <code>e(N)</code>             | number of observations                                   |
| <code>e(N_g)</code>           | number of groups                                         |
| <code>e(N_selected)</code>    | number of selected observations                          |
| <code>e(N_nonselected)</code> | number of nonselected observations                       |
| <code>e(k)</code>             | number of parameters                                     |
| <code>e(k_cat#)</code>        | number of categories for the #th <i>depvar</i> , ordinal |
| <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(k_aux)</code>         | number of auxiliary parameters                           |
| <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(n_quad)</code>        | number of integration points for multivariate normal     |
| <code>e(n_quad3)</code>       | number of integration points for trivariate normal       |
| <code>e(n_requad)</code>      | number of integration points for random effects          |
| <code>e(g_min)</code>         | smallest group size                                      |
| <code>e(g_avg)</code>         | average group size                                       |
| <code>e(g_max)</code>         | largest group size                                       |
| <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>            | xteoprobit                                                                           |
| <code>e(cmdline)</code>        | command as typed                                                                     |
| <code>e(depvar)</code>         | names of dependent variables                                                         |
| <code>e(tsel_ll)</code>        | left-censoring limit for tobit selection                                             |
| <code>e(tsel_ul)</code>        | right-censoring limit for tobit selection                                            |
| <code>e(ivar)</code>           | variable denoting groups                                                             |
| <code>e(title)</code>          | title in estimation output                                                           |
| <code>e(clustvar)</code>       | name of cluster variable                                                             |
| <code>e(offset#)</code>        | offset for the #th <i>depvar</i> , where # is determined by equation order in output |
| <code>e(chi2type)</code>       | Wald; type of model $\chi^2$ 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(reintmethod)</code>    | integration method for random effects                                                |
| <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>     | b V                                                                                  |
| <code>e(estat_cmd)</code>      | program used to implement <code>estat</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(cat#)</code>     | categories for the #th <i>depvar</i> , ordinal |
| <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                      |

In addition to the above, the following is stored in `r()`:

|                       |                                                                                                                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------|
| Matrices              |                                                                                                                       |
| <code>r(table)</code> | matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals |

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

## Methods and formulas

The methods and formulas presented here are for the ordered probit model. The estimators implemented in `eoprobit` and `xteoprobit` are maximum likelihood estimators covered by the results in chapter 13 of [Wooldridge \(2010\)](#) and [White \(1996\)](#).

The log-likelihood functions maximized by `eoprobit` and `xteoprobit` are implied by the triangular structure of the model. Specifically, the joint distribution of the endogenous variables is a product of conditional and marginal distributions because the model is triangular. For a few of the many relevant applications of this result in literature, see chapter 10 of [Amemiya \(1985\)](#); [Heckman \(1976, 1979\)](#); chapter 5 of [Maddala \(1983\)](#); [Maddala and Lee \(1976\)](#); sections 15.7.2, 15.7.3, 16.3.3, 17.5.2, and 19.7.1 in [Wooldridge \(2010\)](#); and [Wooldridge \(2014\)](#). [Roodman \(2011\)](#) and [Bartus and Roodman \(2014\)](#) used this result to derive the formulas discussed below.

Methods and formulas are presented under the following headings:

- Introduction*
- Endogenous covariates*
  - Continuous endogenous covariates*
  - Binary and ordinal endogenous covariates*
- Treatment*
- Endogenous sample selection*
  - Probit endogenous sample selection*
  - Tobit endogenous sample selection*
- Random effects*
- Combinations of features*
- Confidence intervals*

## Introduction

An ordered probit regression of outcome  $y_i$  on covariates  $\mathbf{x}_i$  may be written as

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i \leq \kappa_h$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The unobserved error  $\epsilon_i$  is standard normal.

The log likelihood for this model is

$$\ln L = \sum_{i=1}^N w_i \ln \left[ \begin{array}{l} 1(y_i = v_1) \Phi(-\mathbf{x}_i \boldsymbol{\beta}) \\ + \sum_{h=2}^{H-1} 1(y_i = v_h) \{ \Phi(\kappa_h - \mathbf{x}_i \boldsymbol{\beta}) - \Phi(\kappa_{h-1} - \mathbf{x}_i \boldsymbol{\beta}) \} \\ + 1(y_i = v_H) \Phi(\mathbf{x}_i \boldsymbol{\beta}) \end{array} \right]$$

where  $w_i$  are the weights.

For  $h = 0, \dots, H$ , define

$$c_{ih} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i \boldsymbol{\beta} & h = 1, \dots, H-1 \\ \infty & h = H \end{cases} \quad (1)$$

This leads to the limits

$$l_{1i} = c_{i(h-1)} \quad \text{if } y_i = v_h \quad (2)$$

and

$$u_{1i} = c_{ih} \quad \text{if } y_i = v_h \quad (3)$$

These are limits on the unobserved  $\epsilon_i$  based on the observed values of  $y_i$  and  $\mathbf{x}_i$ . They let us rewrite the log likelihood concisely as

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_1^*(l_{1i}, u_{1i}, 1)$$

The conditional probabilities of success can be written using similar notation. For  $h = 1, \dots, H$ ,

$$\Pr(y_i = v_h | \mathbf{x}_i) = \Phi_1^*(c_{i(h-1)}, c_{ih}, 1) \quad (4)$$

If you are willing to take our word for some derivations and notation, the following is complete. Longer explanations and derivations for some terms and functions are provided in [Methods and formulas](#) of [ERM] **eoprobit**. For example, we need the two-sided probability function  $\Phi_d^*$  that is discussed in [Introduction](#) in [ERM] **eoprobit**.

If you are interested in all the details, we suggest you read [Methods and formulas](#) of [ERM] **eoprobit** in its entirety before reading this section. Here we mainly show how the complications that arise in ERMs are handled in an ordered probit framework.

## Endogenous covariates

### Continuous endogenous covariates

An ordered probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $C$  continuous endogenous covariates  $\mathbf{w}_{ci}$  has the form

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i\boldsymbol{\beta} + \mathbf{w}_{ci}\boldsymbol{\beta}_c + \epsilon_i \leq \kappa_h$$

$$\mathbf{w}_{ci} = \mathbf{z}_{ci}\mathbf{A}_c + \boldsymbol{\epsilon}_{ci}$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The vector  $\mathbf{z}_{ci}$  contains variables from  $\mathbf{x}_i$  and other covariates that affect  $\mathbf{w}_{ci}$ . The unobserved errors  $\epsilon_i$  and  $\boldsymbol{\epsilon}_{ci}$  are multivariate normal with mean 0 and covariance

$$\begin{bmatrix} 1 & \boldsymbol{\sigma}'_{1c} \\ \boldsymbol{\sigma}_{1c} & \boldsymbol{\Sigma}_c \end{bmatrix}$$

As in *Continuous endogenous covariates* in [ERM] **eprobit**, the likelihood can be written using the conditional density of  $\epsilon_i$  on  $\mathbf{w}_{ci}$ .

Now, for  $h = 0, \dots, H$ , define

$$c_{ih} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i\boldsymbol{\beta} - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}(\mathbf{w}_{ci} - \mathbf{z}_{ci}\mathbf{A}_c)' & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

These expressions used the conditional mean of  $\epsilon_i$ . The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = c_{i(h-1)} \quad \text{if} \quad y_i = v_h$$

and

$$u_{1i} = c_{ih} \quad \text{if} \quad y_i = v_h$$

Using these limits, the conditional variance, and the conditional density of  $\mathbf{w}_{ci}$ , we obtain the log likelihood

$$\ln L = \sum_{i=1}^N w_i \{ \ln \Phi_1^*(l_{1i}, u_{1i}, 1 - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}\boldsymbol{\sigma}_{1c}) + \ln \phi_C(\mathbf{w}_{ci} - \mathbf{z}_{ci}\mathbf{A}_c, \boldsymbol{\Sigma}_c) \}$$

The conditional probabilities of success can be written using similar notation. For  $h = 1, \dots, H$ ,

$$\Pr(y_i = v_h | \mathbf{x}_i) = \Phi_1^*(c_{i(h-1)}, c_{ih}, 1 - \boldsymbol{\sigma}'_{1c}\boldsymbol{\Sigma}_c^{-1}\boldsymbol{\sigma}_{1c})$$

## Binary and ordinal endogenous covariates

Here we begin by formulating the ordered probit regression of  $y_i$  on exogenous covariates  $\mathbf{x}_i$  and  $B$  binary and ordinal endogenous covariates  $\mathbf{w}_{bi} = [w_{b1i}, \dots, w_{bBi}]$ . Indicator (dummy) variables for the levels of each binary and ordinal covariate are used in the model. You can also interact other covariates with the binary and ordinal endogenous covariates, as in treatment-effect models.

The binary and ordinal endogenous covariates  $\mathbf{w}_{bi}$  are formulated as in [Binary and ordinal endogenous covariates](#) in [ERM] **eoprobit**.

The model for the outcome can be formulated with or without different correlation parameters for each level of  $\mathbf{w}_{bi}$ . Level-specific parameters are obtained by specifying `pocorrelation` in the `endogenous()` option.

If the correlation parameters are not level specific, we have

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i\boldsymbol{\beta} + \mathbf{wind}_{b1i}\boldsymbol{\beta}_{b1} + \dots + \mathbf{wind}_{bBi}\boldsymbol{\beta}_{bB} + \epsilon_i \leq \kappa_h$$

where the values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The  $\mathbf{wind}_{bj}$  vectors are defined in [Binary and ordinal endogenous covariates](#) in [ERM] **eoprobit**. The outcome error  $\epsilon_i$  and binary and ordinal endogenous errors  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  are multivariate normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \boldsymbol{\rho}'_{1b} \\ \boldsymbol{\rho}_{1b} & \boldsymbol{\Sigma}_b \end{bmatrix}$$

From here, we discuss the model with ordinal endogenous covariates. The results for binary endogenous covariates are similar.

Now, for  $h = 0, \dots, H$ , define

$$c_{ih} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i\boldsymbol{\beta} - \mathbf{wind}_{b1i}\boldsymbol{\beta}_{b1} - \dots - \mathbf{wind}_{bBi}\boldsymbol{\beta}_{bB} & h = 1, \dots, H - 1 \\ \infty & h = H \end{cases}$$

The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = c_{i(h-1)} \quad \text{if} \quad y_i = v_h$$

and

$$u_{1i} = c_{ih} \quad \text{if} \quad y_i = v_h$$

Let

$$\mathbf{l}_i = [l_{1i} \quad l_{b1i} \quad \dots \quad l_{bBi}]$$

$$\mathbf{u}_i = [u_{1i} \quad u_{b1i} \quad \dots \quad u_{bBi}]$$

where the  $l_{bj}$  and  $u_{bj}$  are the lower and upper limits for the binary and ordinal endogenous covariate probabilities. They are defined in [Binary and ordinal endogenous covariates](#) in [ERM] **eoprobit**.

So the log likelihood for this model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_{B+1}^*(\mathbf{l}_i, \mathbf{u}_i, \boldsymbol{\Sigma})$$

Now let

$$\begin{aligned}\mathbf{l}_{bi} &= [l_{b1i} \ \dots \ l_{bBi}] \\ \mathbf{u}_{bi} &= [u_{b1i} \ \dots \ u_{bBi}] \\ \mathbf{l}_{ih1} &= [c_{i(h-1)} \ \mathbf{l}_{bi}] \\ \mathbf{u}_{ih1} &= [c_{ih} \ \mathbf{u}_{bi}]\end{aligned}$$

The conditional probabilities are

$$\Pr(y_i = v_h | \mathbf{x}_i, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{w}_{bi}) = \frac{\Phi_{B+1}^*(\mathbf{l}_{ih1}, \mathbf{u}_{ih1}, \boldsymbol{\Sigma})}{\Phi_B^*(\mathbf{l}_{bi}, \mathbf{u}_{bi}, \boldsymbol{\Sigma}_b)}$$

When the endogenous ordinal variables are different treatments, holding the correlation parameters constant over the treatment levels is a constrained form of the potential-outcome model. In an unconstrained potential-outcome model, the correlations between the outcome and the treatments—the endogenous ordinal regressors  $\mathbf{w}_{bi}$ —vary over the levels of each treatment.

In this unconstrained model, there is a different potential-outcome error for each level of each treatment. For example, when the endogenous treatment variable  $w_1$  has three levels (0, 1, and 2) and the endogenous treatment variable  $w_2$  has four levels (0, 1, 2, and 3), the unconstrained model has  $12 = 3 \times 4$  outcome errors. Because there is a different correlation between each potential outcome and each endogenous treatment, there are  $2 \times 12$  correlation parameters between the potential outcomes and the treatments in this example model.

We denote the number of different combinations of values for the endogenous treatments  $\mathbf{w}_{bi}$  by  $M$ , and we denote the vector of values in each combination by  $\mathbf{v}_j$  ( $j \in \{1, 2, \dots, M\}$ ). Letting  $k_{wp}$  be the number of levels of endogenous ordinal treatment variable  $p \in \{1, 2, \dots, B\}$  implies that  $M = k_{w1} \times k_{w2} \times \dots \times k_{wB}$ .

In this case, we have

$$y_i = \sum_{j=1}^M 1(\mathbf{w}_{bi} = \mathbf{v}_j) y_{ji}$$

where for  $j = 1, \dots, M$ ,

$$y_{ji} = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i \boldsymbol{\beta} + \mathbf{w}_{b1i} \boldsymbol{\beta}_{b1} + \dots + \mathbf{w}_{bBi} \boldsymbol{\beta}_{bB} + \epsilon_{ji} \leq \kappa_h$$

The outcome errors  $\epsilon_{ji}$  and the endogenous errors  $\epsilon_{b1i}, \dots, \epsilon_{bBi}$  are multivariate normal with 0 mean and covariance

$$\boldsymbol{\Sigma}_j = \begin{bmatrix} 1 & \rho'_{j1b} \\ \rho_{j1b} & \boldsymbol{\Sigma}_b \end{bmatrix}$$

Now let

$$\boldsymbol{\Sigma}_{i,b} = \sum_{j=1}^M 1(\mathbf{w}_{bi} = \mathbf{v}_j) \boldsymbol{\Sigma}_j$$

Now the log likelihood for this model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_{B+1}^*(\mathbf{l}_i, \mathbf{u}_i, \boldsymbol{\Sigma}_{i,b})$$

The conditional probabilities are

$$\Pr(y_i = v_h | \mathbf{x}_i, \mathbf{z}_{b1i}, \dots, \mathbf{z}_{bBi}, \mathbf{w}_{bi}) = \frac{\Phi_{B+1}^*(\mathbf{l}_{ih1}, \mathbf{u}_{ih1}, \boldsymbol{\Sigma}_{i,b})}{\Phi_B^*(\mathbf{l}_{bi}, \mathbf{u}_{bi}, \boldsymbol{\Sigma}_b)}$$

## Treatment

In the potential-outcomes framework, the treatment  $t_i$  is a discrete variable taking  $T$  values, indexing the  $T$  potential outcomes of the outcome  $y_i$ :  $y_{1i}, \dots, y_{Ti}$ .

When we observe treatment  $t_i$  with levels  $v_1, \dots, v_T$ , we have

$$y_i = \sum_{j=1}^T 1(t_i = v_{tj})y_{ji}$$

So for each observation, we observe only the potential outcome associated with that observation's treatment value.

For exogenous treatments, our approach is equivalent to the regression adjustment treatment-effect estimation method. See [CAUSAL] **teffects intro advanced**. We do not model the treatment assignment process. The formulas for the treatment effects and potential-outcome means (POMs) are equivalent to what we provide here for endogenous treatments. The treatment effect on the treated for  $\mathbf{x}_i$  for an exogenous treatment is equivalent to what we provide here for the endogenous treatment when the correlation parameter between the outcome and treatment errors is set to 0. The average treatment effects (ATEs) and POMs for exogenous treatments are estimated as predictive margins in an analogous manner to what we describe here for endogenous treatments.

From here, we assume an endogenous treatment  $t_i$ . As in *Treatment* in [ERM] **eoprobit**, we model the treatment assignment process with a probit or an ordered probit model, and we call the treatment assignment error  $\epsilon_{ti}$ . An ordered probit regression of  $y_i$  on treatment  $t_i$  with levels  $v_{t1}, \dots, v_{tT}$  has the form

$$y_i = \sum_{j=1}^T 1(t_i = v_{tj})y_{ji}$$

where for  $j = 1, \dots, T$  and exogenous covariates  $\mathbf{x}_i$

$$y_{ji} = v_h \quad \text{iff} \quad \kappa_{(h-1)j} < \mathbf{x}_i\boldsymbol{\beta}_j + \epsilon_{ji} \leq \kappa_{hj}$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ . For  $j = 1, \dots, T$ ,  $\kappa_{0j}$  is taken as  $-\infty$  and  $\kappa_{Hj}$  is taken as  $+\infty$ .

This model can be formulated with or without different correlation parameters for each potential outcome. Potential-outcome specific parameters are obtained by specifying `pocorrelation` in the `entreat()` option.

If the correlation parameters are not potential-outcome specific, for  $j = 1, \dots, T$ ,  $\epsilon_{ji}$  and  $\epsilon_{ti}$  are bivariate normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{1t} \\ \rho_{1t} & 1 \end{bmatrix}$$

The treatment is exogenous if  $\rho_{1t} = 0$ . Note that we did not specify the structure of the correlations between the potential-outcome errors. We do not need information about these correlations to estimate POMs and treatment effects because all covariates and the outcome are observed in observations from each group.

From here, we discuss a model with an ordinal endogenous treatment. The results for binary treatment models are similar. Because the unobserved errors are bivariate normal, we can express the log likelihood in terms of the  $\Phi_2^*$  function.

For  $j = 1, \dots, T$  and  $h = 0, \dots, H$ , let

$$c_{1ihj} = \begin{cases} -\infty & h = 0 \\ \kappa_{hj} - \mathbf{x}_i \boldsymbol{\beta}_j & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

The lower and upper limits for the  $y_i$  probability are

$$l_{1i} = c_{i(h-1)j} \quad \text{if } y_i = v_h, t_i = v_{tj}$$

and

$$u_{1i} = c_{ihj} \quad \text{if } y_i = v_h, t_i = v_{tj}$$

The log likelihood for the model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_2^*([l_{1i} \quad l_{ti}], [u_{1i} \quad u_{ti}], \boldsymbol{\Sigma})$$

where the lower and upper limits for the treatment probability,  $l_{ti}$  and  $u_{ti}$ , are defined in [Treatment](#) in [\[ERM\] eprobit](#).

The conditional probability of obtaining treatment level  $v_{th}$  is

$$\Pr(t_i = v_{th} | \mathbf{z}_{ti}) = \Phi_1^*(c_{ti(h-1)}, c_{tih}, 1)$$

where the cutpoints for the treatment probabilities  $c_{tij}$  are defined in [Treatment](#) in [\[ERM\] eprobit](#).

For  $h = 1, \dots, H$ , the conditional probabilities for outcome level  $v_h$  at treatment level  $v_{tj}$  are

$$\Pr(y_i = v_h | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_{tj}) = \frac{\Phi_2^*([c_{1i(h-1)j} \quad c_{ti(j-1)}], [c_{1ihj} \quad c_{tij}], \boldsymbol{\Sigma})}{\Phi_1^*(c_{ti(j-1)}, c_{tij}, 1)}$$

The conditional POM for treatment group  $j$  and outcome category  $h$  is

$$\text{POM}_{hj}(\mathbf{x}_i) = E \{1(y_{ji} = v_h) | \mathbf{x}_i\} = \Phi_1^*(c_{1i(h-1)j}, c_{1i(h-1)j}, 1)$$

Conditional on the covariates  $\mathbf{x}_i$  and  $\mathbf{z}_{ti}$  and the treatment  $t_i = v_m$ , the POM for treatment group  $j$  and outcome category  $h$  is

$$\begin{aligned} \text{POM}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) &= E \{1(y_{ji} = v_h) | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m\} \\ &= \frac{\Phi_2^*([c_{1i(h-1)j} \quad c_{ti(m-1)}], [c_{1ihj} \quad c_{tim}], \boldsymbol{\Sigma})}{\Phi_1^*(c_{ti(m-1)}, c_{tim}, 1)} \end{aligned}$$

Without loss of generality,  $t_i = v_{t1}$  corresponds to the control or base level of the treatment. Treatment effects are the differences between the potential outcomes  $y_{2i}, \dots, y_{Ti}$  and the control  $y_{1i}$ . When the potential outcomes are ordered probit, the treatment effect on a particular category is of interest.



The treatment effect of treatment group  $j$  on category  $h$  is  $1(y_{ji} = v_h) - 1(y_{1i} = v_h)$ , the difference in the outcome for individual  $i$  on being in category  $h$  if the individual receives the treatment  $t_i = v_{tj}$  instead of the control  $t_i = v_{t1}$ . Evaluating this treatment effect lets us see how the treatment affects the probability of belonging to outcome category  $h$ .

For treatment group  $j$ , the treatment effect (TE) on category  $h$  conditioned on  $\mathbf{x}_i$  is

$$\begin{aligned} \text{TE}_{hj}(\mathbf{x}_i) &= E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h) | \mathbf{x}_i\} \\ &= \text{POM}_{hj}(\mathbf{x}_i) - \text{POM}_{h1}(\mathbf{x}_i) \end{aligned}$$

For treatment group  $j$ , the treatment effect on the treated (TET) on category  $h$  in treatment group  $m$  conditioned on  $\mathbf{x}_i$  and  $\mathbf{z}_{ti}$  is

$$\begin{aligned} \text{TET}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) &= E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h) | \mathbf{x}_i, t_i = v_{t,m}\} \\ &= \text{POM}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) - \text{POM}_{h1}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) \end{aligned}$$

We can take the expectation of these conditional predictions over the covariates to get population average parameters. The `estat teffects` or `margins` command is used to estimate the expectations as predictive margins once the model is fit with `eoprobit`. The POM for treatment group  $j$  and outcome category  $h$  is

$$\text{POM}_{hj} = E \{1(y_{ji} = v_h)\} = E \{\text{POM}_{hj}(\mathbf{x}_i)\}$$

The ATE for treatment group  $j$  and outcome category  $h$  is

$$\text{ATE}_{hj} = E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h)\} = E \{\text{TE}_{hj}(\mathbf{x}_i)\}$$

For treatment group  $j$ , the average treatment effect on the treated (ATET) for outcome category  $h$  in treatment group  $m$  is

$$\begin{aligned} \text{ATET}_{hjm} &= E \{1(y_{ji} = v_h) - 1(y_{1i} = v_h) | t_i = v_m\} \\ &= E \{\text{TET}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) | t_i = v_m\} \end{aligned}$$

If the correlation parameters are potential-outcome specific, for  $j = 1, \dots, T$ ,  $\epsilon_{ji}$  and  $\epsilon_{ti}$  are bivariate normal with mean 0 and covariance

$$\Sigma_j = \begin{bmatrix} 1 & \rho_{j1t} \\ \rho_{j1t} & 1 \end{bmatrix}$$

Now define

$$\Sigma_i = \sum_{j=1}^T 1(t_i = v_j) \Sigma_j$$

The log likelihood for the potential-outcome specification correlation model is

$$\ln L = \sum_{i=1}^N w_i \ln \Phi_2^*([l_{1i} \quad l_{ti}], [u_{1i} \quad u_{ti}], \Sigma_i)$$

For  $h = 1, \dots, H$ , the conditional probabilities for outcome level  $v_h$  at treatment level  $v_{t_j}$  are now

$$\Pr(y_i = v_h | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_{t_j}) = \frac{\Phi_2^*([c_{1i(h-1)j} \quad c_{ti(j-1)}], [c_{1ihj} \quad c_{tij}], \boldsymbol{\Sigma}_j)}{\Phi_1^*(c_{ti(j-1)}, c_{tij}, 1)}$$

The conditional POM for exogenous covariates  $\mathbf{x}_i$ , treatment group  $j$ , and outcome category  $h$  has the same definition as in the single correlation case. However, when we also condition on the treatment level  $t_i = v_m$  and  $\mathbf{z}_{ti}$ , the POM for treatment group  $j$  and outcome category  $h$  is

$$\begin{aligned} \text{POM}_{hj}(\mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_m) &= E \{ 1(y_{ij} = v_h) | \mathbf{x}_i, \mathbf{z}_{ti}, t_i = v_{tm} \} \\ &= \frac{\Phi_2^*([c_{1i(h-1)j} \quad c_{ti(m-1)}], [c_{1ihj} \quad c_{tim}], \boldsymbol{\Sigma}_j)}{\Phi_1^*(c_{ti(m-1)}, c_{tim}, 1)} \end{aligned}$$

Treatment effects are formulated as in the single correlation case but using these updated POM definitions. We can take the expectation of these conditional predictions over the covariates to get population-averaged parameters. The `estat teffects` or `margins` command is used to estimate the expectations as predictive margins once the model is fit with `eoprobit`.

## Endogenous sample selection

### Probit endogenous sample selection

An ordered probit model for outcome  $y_i$  with selection on  $s_i$  has the form

$$\begin{aligned} y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i \boldsymbol{\beta} + \epsilon_i \leq \kappa_h \\ s_i = 1 \quad (\mathbf{z}_{si} \boldsymbol{\alpha}_s + \epsilon_{si} > 0) \end{aligned}$$

where  $\mathbf{x}_i$  are covariates that affect the outcome and  $\mathbf{z}_{si}$  are covariates that affect selection. The outcome  $y_i$  is observed if  $s_i = 1$  and is not observed if  $s_i = 0$ . The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ .

The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho_{1s} \\ \rho_{1s} & 1 \end{bmatrix}$$

The lower and upper limits for the  $y_i$  probability,  $l_{1i}$  and  $u_{1i}$ , are as defined in (1)–(3). For the selection indicator, the lower and upper limits  $l_{si}$  and  $u_{si}$  are defined in *Probit endogenous sample selection* in [ERM] `eoprobit`.

The log likelihood for the model is

$$\ln L = \sum_{i \in S} w_i \ln \Phi_2^*([l_{1i} \quad l_{si}], [u_{1i} \quad u_{si}], \boldsymbol{\Sigma}) + \sum_{i \notin S} w_i \ln \Phi_1^*(l_{si}, u_{si}, 1)$$

where  $S$  is the set of observations for which  $y_i$  is observed.

In this model, the probability of success is usually predicted conditional on the covariates  $\mathbf{x}_i$  and not on the selection status  $s_i$ . The formulas for the conditional probability are thus the same as in (4).

The conditional probability of selection is

$$\Pr(s_i = 1 | \mathbf{z}_{si}) = \Phi_1^*(-\mathbf{z}_{si} \boldsymbol{\alpha}_s, \infty, 1)$$

## Tobit endogenous sample selection

Instead of constraining the selection indicator to be binary, tobit endogenous sample selection uses a censored continuous sample-selection indicator. We allow the selection variable to be left- or right-censored.

An ordered probit model for outcome  $y_i$  with tobit selection on  $s_i$  has the form

$$y_i = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i \leq \kappa_h$$

where the values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ .

We observe the selection indicator  $s_i$ , which indicates the censoring status of the latent selection variable  $s_i^*$ ,

$$s_i^* = \mathbf{z}_{si}\boldsymbol{\alpha}_s + \epsilon_{si}$$

$$s_i = \begin{cases} l_i & s_i^* \leq l_i \\ s_i^* & l_i < s_i^* < u_i \\ u_i & s_i^* \geq u_i \end{cases}$$

where  $\mathbf{z}_{si}$  are covariates that affect selection and  $l_i$  and  $u_i$  are fixed lower and upper limits.

The outcome  $y_i$  is observed when  $s_i^*$  is not censored. If  $l_i < s_i^* < u_i$ , then  $y_i$  is observed.  $y_i$  is not observed if  $s_i^* \leq l_i$ , that is, if  $s_i^*$  is left-censored.  $y_i$  is also not observed if  $s_i^*$  is right-censored,  $s_i^* \geq u_i$ . The unobserved errors  $\epsilon_i$  and  $\epsilon_{si}$  are normal with mean 0 and covariance

$$\begin{bmatrix} 1 & \rho_{1s}\sigma_s \\ \rho_{1s}\sigma_s & \sigma_s^2 \end{bmatrix}$$

For the selected observations, we can treat  $s_i$  as a continuous endogenous regressor, as in [Continuous endogenous covariates](#). In fact,  $s_i$  may even be used as a regressor for  $y_i$  in eoprobit (specify `tobitselect(... main)`). On the nonselected observations, we treat  $s_i$  like the probit endogenous sample-selection indicator in [Probit endogenous sample selection](#).

The conditional mean of  $\epsilon_i$  is used in the lower and upper limits for the  $y_i$  probability for selected observations. Let

$$c_{i,h} = \begin{cases} -\infty & h = 0 \\ \kappa_h - \mathbf{x}_i\boldsymbol{\beta} - \rho_{1s}\sigma_s^{-1}(s_i - \mathbf{z}_{si}\boldsymbol{\alpha}_c) & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

The limits for the  $y_i$  probability for selected observations are

$$l_{1i} = c_{i(h-1)} \quad \text{if} \quad y_i = v_h$$

and

$$u_{1i} = c_{ih} \quad \text{if} \quad y_i = v_h$$

It follows that the log likelihood is

$$\begin{aligned} \ln L &= \sum_{i \in S} w_i \{ \ln \Phi_1^*(l_{1i}, u_{1i}, 1 - \rho_{1s}^2) + \ln \phi(s_i - \mathbf{z}_{si} \boldsymbol{\alpha}_s, \sigma_s^2) \} \\ &\quad + \sum_{i \in L} w_i \ln \Phi_1^*(l_{li}, u_{li}, 1) \\ &\quad + \sum_{i \in U} w_i \ln \Phi_1^*(l_{ui}, u_{ui}, 1) \end{aligned}$$

where  $S$  is the set of observations for which  $y_i$  is observed,  $L$  is the set of observations where  $s_i^*$  is left-censored, and  $U$  is the set of observations where  $s_i^*$  is right-censored. The lower and upper limits for selection— $l_{li}$ ,  $u_{li}$ ,  $l_{ui}$ , and  $u_{ui}$ —are defined in [Tobit endogenous sample selection \[ERM\] eoprobit](#).

The conditional probabilities on  $s_i = S_i$  are

$$\Pr(y_i = v_h | \mathbf{x}_i) = \Phi_1^*(c_{i(h-1)}, c_{ih}, 1 - \rho_{1s}^2)$$

If we do not include  $s_i$  in the main outcome equation, the probability of success is calculated as (4) again.

## Random effects

For an ordered probit regression with random effects, we observe panel data. For panel  $i = 1, \dots, N$  and observation  $j = 1, \dots, N_i$ , an ordered probit regression of  $y_{ij}$  on covariates  $\mathbf{x}_{ij}$  with random effect  $u_i$  may be written as

$$y_{ij} = v_h \quad \text{iff} \quad \kappa_{h-1} < \mathbf{x}_{ij} \boldsymbol{\beta} + \epsilon_{ij} + u_i \leq \kappa_h$$

The values  $v_1, \dots, v_H$  are real numbers such that  $v_h < v_m$  for  $h < m$ .  $\kappa_0$  is taken as  $-\infty$  and  $\kappa_H$  is taken as  $+\infty$ . The random effect  $u_i$  is normal with mean 0 and variance  $\sigma_u^2$ . It is independent of the observation-level error  $\epsilon_{ij}$ , which is standard normal.

We derive the likelihood by using the conditional density of  $y_{ij}$  on the random effect  $u_i$  and the marginal density of  $u_i$ . Multiplying them together, we have the joint density, which is integrated over  $u_i$ .

Let

$$l_{ij}(u) = \left[ \begin{aligned} &1(y_{ij} = v_1) \Phi(-\mathbf{x}_{ij} \boldsymbol{\beta} - u) \\ &+ \sum_{h=2}^{H-1} 1(y_{ij} = v_h) \{ \Phi(\kappa_h - \mathbf{x}_{ij} \boldsymbol{\beta} - u) - \Phi(\kappa_{h-1} - \mathbf{x}_{ij} \boldsymbol{\beta} - u) \} \\ &+ 1(y_{ij} = v_H) \Phi(\mathbf{x}_{ij} \boldsymbol{\beta} + u) \end{aligned} \right]$$

The likelihood for panel  $i$  is

$$L_i = \int_{-\infty}^{\infty} \phi\left(\frac{u_i}{\sigma_u}\right) \prod_{j=1}^{N_i} l_{ij}(u_i) du_i$$

We can approximate this integral using Gauss–Hermite quadrature. For  $q$ -point Gauss–Hermite quadrature, let the abscissa and weight pairs be denoted by  $(a_{ki}, w_{ki})$ ,  $k = 1, \dots, q$ . The Gauss–Hermite quadrature approximation is then

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2) dx \approx \sum_{k=1}^q w_{ki} f(a_{ki})$$

The default approximation used by `xteoprobit` is mean–variance adaptive Gauss–Hermite quadrature. This chooses optimal abscissa and weights for each panel. See [Likelihood for multiequation models](#) in [ERM] `eoprobit` for more information on the use of mean–variance adaptive Gauss–Hermite quadrature.

Using the quadrature approximation, the log likelihood is

$$\ln L = \sum_{i=1}^N \ln \left\{ \sum_{k=1}^q w_{ki} \prod_{j=1}^{N_i} l_{ij}(\sigma_u a_{ki}) \right\}$$

Now we will derive the conditional probabilities of success. These are similar to those given in [Introduction](#), but the variance input to  $\Phi_1^*$  is the variance of the random effect plus the observation-level error.

First, let

$$\xi_{ij} = \epsilon_{ij} + u_i$$

$\xi_{ij}$  is normal with mean 0 and variance  $\sigma_\xi^2 = 1 + \sigma_u^2$ .

Now, for  $h = 0, \dots, H$ , define

$$c_{ijh} = \begin{cases} -\infty & h = 0 \\ (\kappa_h - \mathbf{x}_{ij}\beta) & h = 1, \dots, H-1 \\ \infty & h = H \end{cases}$$

For  $h = 1, \dots, H$ , the conditional probabilities of success are

$$\Pr(y_{ij} = v_h | \mathbf{x}_{ij}) = \Phi_1^*(c_{ij(h-1)}, c_{ijh}, \sigma_\xi^2)$$

## Combinations of features

Extended ordered probit regression models that involve multiple features can be formulated using the techniques discussed in [Likelihood for multiequation models](#) in [ERM] `eoprobit`. Essentially, the density of the observed endogenous covariates can be written in terms of the unobserved normal errors. The observed endogenous and exogenous covariates determine the range of the errors, and the joint density can be evaluated as multivariate normal probabilities and densities.

## Confidence intervals

The estimated variances will always be nonnegative, and the estimated correlations will always fall in  $(-1, 1)$ . We use transformations to obtain confidence intervals that accommodate these ranges.

We use the log transformation to obtain the confidence intervals for variance parameters and the atanh transformation to obtain confidence intervals for correlation parameters. For details, see *Confidence intervals* in [ERM] **eoprobit**.

## References

- Amemiya, T. 1985. *Advanced Econometrics*. Cambridge, MA: Harvard University Press.
- Bartus, T., and D. Roodman. 2014. Estimation of multiprocess survival models with `cmp`. *Stata Journal* 14: 756–777.
- Botezat, A., and F. Pfeiffer. 2014. The impact of parents' migration on the well-being of children left behind: Initial evidence from Romania. IZA Discussion Paper No. 8225, Institute for the Study of Labor (IZA). <http://ftp.iza.org/dp8225.pdf>.
- Chyi, H., and S. Mao. 2012. The determinants of happiness of China's elderly population. *Journal of Happiness Studies* 13: 167–185. <https://doi.org/10.1007/s10902-011-9256-8>.
- De Luca, G., and V. Perotti. 2011. Estimation of ordered response models with sample selection. *Stata Journal* 11: 213–239.
- Gregory, C. A. 2015. Estimating treatment effects for ordered outcomes using maximum simulated likelihood. *Stata Journal* 15: 756–774.
- Heckman, J. J. 1976. The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models. *Annals of Economic and Social Measurement* 5: 475–492.
- . 1979. Sample selection bias as a specification error. *Econometrica* 47: 153–161. <https://doi.org/10.2307/1912352>.
- Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press.
- Maddala, G. S., and L.-F. Lee. 1976. Recursive models with qualitative endogenous variables. *Annals of Economic and Social Measurement* 5: 525–545.
- Roodman, D. 2011. Fitting fully observed recursive mixed-process models with `cmp`. *Stata Journal* 11: 159–206.
- White, H. L., Jr. 1996. *Estimation, Inference and Specification Analysis*. Cambridge: Cambridge University Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.
- . 2014. Quasi-maximum likelihood estimation and testing for nonlinear models with endogenous explanatory variables. *Journal of Econometrics* 182: 226–234. <https://doi.org/10.1016/j.jeconom.2014.04.020>.

## Also see

- [ERM] **eoprobit postestimation** — Postestimation tools for `eoprobit` and `xteoprobit`
- [ERM] **eoprobit predict** — `predict` after `eoprobit` and `xteoprobit`
- [ERM] **predict advanced** — `predict`'s advanced features
- [ERM] **predict treatment** — `predict` for treatment statistics
- [ERM] **estat teffects** — Average treatment effects for extended regression models
- [ERM] **Intro 9** — Conceptual introduction via worked example
- [R] **heckoprobit** — Ordered probit model with sample selection
- [R] **oprobit** — Ordered probit regression
- [SVY] **svy estimation** — Estimation commands for survey data
- [XT] **xteoprobit** — Random-effects ordered probit models

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

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

