

oprobit — Ordered probit regression

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

Description

`oprobit` fits ordered probit models of ordinal variable *depvar* on the independent variables *indepvars*. The actual values taken on by the dependent variable are irrelevant, except that larger values are assumed to correspond to “higher” outcomes.

Quick start

Ordinal probit model of *y* on *x1* and categorical variables *a* and *b*

```
oprobit y x1 i.a i.b
```

Model of *y* on *x1* and a one-period lagged value of *x1* using `tsset` data

```
oprobit y x1 L.x1
```

Same as above, but calculate results for each level of *catvar* and save statistics to *myfile.dta*

```
statsby, by(catvar) saving(myfile): oprobit y x1 L.x1
```

Menu

Statistics > Ordinal outcomes > Ordered probit regression

Syntax

```
oprobit deivar [indepvars] [if] [in] [weight] [, options]
```

<i>options</i>	Description
Model	
<code>offset(<i>varname</i>)</code>	include <i>varname</i> in model with coefficient constrained to 1
<code>constraints(<i>constraints</i>)</code>	apply specified linear constraints
SE/Robust	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>oim</code> , <code>opg</code> , <code>robust</code> , <code>cluster <i>clustvar</i></code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>nocnsreport</code>	do not display constraints
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>collinear</code>	keep collinear variables
<code>coeflegend</code>	display legend instead of statistics
<i>indepvars</i> may contain factor variables; see [U] 11.4.3 Factor variables .	
<i>deivar</i> and <i>indepvars</i> may contain time-series operators; see [U] 11.4.4 Time-series varlists .	
<code>bayes</code> , <code>bootstrap</code> , <code>by</code> , <code>collect</code> , <code>fmm</code> , <code>fp</code> , <code>jackknife</code> , <code>mfp</code> , <code>mi estimate</code> , <code>nestreg</code> , <code>rolling</code> , <code>statsby</code> , <code>stepwise</code> , and <code>svy</code> are allowed; see [U] 11.1.10 Prefix commands . For more details, see [BAYES] bayes: oprobit and [FMM] fmm: oprobit .	
<code>vce(bootstrap)</code> and <code>vce(jackknife)</code> are not allowed with the <code>mi estimate</code> prefix; see [MI] mi estimate .	
Weights are not allowed with the <code>bootstrap</code> prefix; see [R] bootstrap .	
<code>vce()</code> and weights are not allowed with the <code>svy</code> prefix; see [SVY] svy .	
<code>fweights</code> , <code>iweights</code> , and <code>pweights</code> are allowed; see [U] 11.1.6 weight .	
<code>collinear</code> and <code>coeflegend</code> do not appear in the dialog box.	
See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.	

Options

Model

`offset(varname)`, `constraints(constraints)`; 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`: [noci](#), [nopvalues](#), [noomitted](#), [vsquish](#), [noemptycells](#), [baselevels](#), [allbaselevels](#), [nofvlabel](#), [fvwrap\(#\)](#), [fvwrapon\(style\)](#), [cformat\(%fmt\)](#), [pformat\(%fmt\)](#), [sformat\(%fmt\)](#), and [nolstretch](#); see [R] [Estimation options](#).

Maximization

`maximize_options`: [difficult](#), [technique\(algorithm_spec\)](#), [iterate\(#\)](#), [\[no\]log](#), [trace](#), [gradient](#), [showstep](#), [hessian](#), [showtolerance](#), [tolerance\(#\)](#), [ltolerance\(#\)](#), [nrtolerance\(#\)](#), [nonrto](#), and [from\(init_specs\)](#); see [R] [Maximize](#). These options are seldom used.

The following options are available with `oprobit` but is not shown in the dialog box:

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

Remarks and examples

stata.com

An ordered probit model is used to estimate relationships between an ordinal dependent variable and a set of independent variables. An *ordinal* variable is a variable that is categorical and ordered, for instance, “poor”, “good”, and “excellent”, which might indicate a person’s current health status or the repair record of a car. If there are only two outcomes, see [R] [logistic](#), [R] [logit](#), and [R] [probit](#). This entry is concerned only with more than two outcomes. If the outcomes cannot be ordered (for example, residency in the north, east, south, or west), see [R] [mlogit](#). This entry is concerned only with models in which the outcomes can be ordered. See [R] [logistic](#) for a list of related estimation commands.

In ordered probit, an underlying score is estimated as a linear function of the independent variables and a set of cutpoints. The probability of observing outcome i corresponds to the probability that the estimated linear function, plus random error, is within the range of the cutpoints estimated for the outcome:

$$\Pr(\text{outcome}_j = i) = \Pr(\kappa_{i-1} < \beta_1 x_{1j} + \beta_2 x_{2j} + \cdots + \beta_k x_{kj} + u_j \leq \kappa_i)$$

u_j is assumed to be normally distributed. In either case, we estimate the coefficients $\beta_1, \beta_2, \dots, \beta_k$ together with the cutpoints $\kappa_1, \kappa_2, \dots, \kappa_{I-1}$, where I is the number of possible outcomes. κ_0 is taken as $-\infty$, and κ_I is taken as $+\infty$. All of this is a direct generalization of the ordinary two-outcome probit model.

► Example 1

In [example 2](#) of [R] [ologit](#), we use a variation of the automobile dataset (see [U] [1.2.2 Example datasets](#)) to analyze the 1977 repair records of 66 foreign and domestic cars. We use ordered logit to explore the relationship of `rep77` in terms of `foreign` (origin of manufacture), `length` (a proxy for size), and `mpg`. Here we fit the same model using ordered probit rather than ordered logit:

```

. use https://www.stata-press.com/data/r18/fullauto
(Automobile models)

. oprobit rep77 foreign length mpg

Iteration 0: Log likelihood = -89.895098
Iteration 1: Log likelihood = -78.106316
Iteration 2: Log likelihood = -78.020086
Iteration 3: Log likelihood = -78.020025
Iteration 4: Log likelihood = -78.020025

Ordered probit regression

Log likelihood = -78.020025

Number of obs = 66
LR chi2(3) = 23.75
Prob > chi2 = 0.0000
Pseudo R2 = 0.1321

```

rep77	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
foreign	1.704861	.4246796	4.01	0.000	.8725037	2.537217
length	.0468675	.012648	3.71	0.000	.022078	.0716571
mpg	.1304559	.0378628	3.45	0.001	.0562463	.2046656
/cut1	10.1589	3.076754			4.128577	16.18923
/cut2	11.21003	3.107527			5.119389	17.30067
/cut3	12.54561	3.155233			6.361467	18.72975
/cut4	13.98059	3.218793			7.671874	20.28931

We find that foreign cars have better repair records, as do larger cars and cars with better mileage ratings.

◀

Stored results

oprobit stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_cdd)</code>	number of completely determined observations
<code>e(k_cat)</code>	number of categories
<code>e(k)</code>	number of parameters
<code>e(k_aux)</code>	number of auxiliary 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_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(r2_p)</code>	pseudo- R^2
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model
<code>e(N_clust)</code>	number of clusters
<code>e(chi2)</code>	χ^2
<code>e(p)</code>	p -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>	oprobit
<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(offset)</code>	linear offset variable
<code>e(chi2type)</code>	Wald or LR; type of model χ^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(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(predict)</code>	program used to implement <code>predict</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
------------------------	-------------------------

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.

Methods and formulas

See *Methods and formulas* of [R] `ologit`.

References

- Aitchison, J., and S. D. Silvey. 1957. The generalization of probit analysis to the case of multiple responses. *Biometrika* 44: 131–140. <https://doi.org/10.2307/2333245>.
- Bauldry, S., J. Xu, and A. S. Fullerton. 2018. `gencrm`: A new command for generalized continuation-ratio models. *Stata Journal* 18: 924–936.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Canette, I. 2013. Fitting ordered probit models with endogenous covariates with Stata’s `gsem` command. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2013/11/07/fitting-ordered-probit-models-with-endogenous-covariates-with-statas-gsem-command/>.
- Chiburis, R., and M. Lokshin. 2007. Maximum likelihood and two-step estimation of an ordered-probit selection model. *Stata Journal* 7: 167–182.
- De Luca, G., and V. Perotti. 2011. Estimation of ordered response models with sample selection. *Stata Journal* 11: 213–239.

- Drukker, D. M. 2016. An ordered-probit inverse probability weighted (IPW) estimator. *The Stata Blog: Not Elsewhere Classified*. <http://blog.stata.com/2016/09/13/an-ordered-probit-inverse-probability-weighted-ipw-estimator/>.
- Huismans, J., J. W. Nijenhuis, and A. Sirchenko. 2022. A mixture of ordered probit models with endogenous switching between two latent classes. *Stata Journal* 22: 557–596.
- Long, J. S. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage.
- Long, J. S., and J. Freese. 2014. *Regression Models for Categorical Dependent Variables Using Stata*. 3rd ed. College Station, TX: Stata Press.
- Miranda, A., and S. Rabe-Hesketh. 2006. Maximum likelihood estimation of endogenous switching and sample selection models for binary, ordinal, and count variables. *Stata Journal* 6: 285–308.
- Smith, E. K., M. G. Lacy, and A. Mayer. 2019. Performance simulations for categorical mediation: Analyzing khb estimates of mediation in ordinal regression models. *Stata Journal* 19: 913–930.
- Stewart, M. B. 2004. Semi-nonparametric estimation of extended ordered probit models. *Stata Journal* 4: 27–39.
- Williams, R. 2010. Fitting heterogeneous choice models with `oglm`. *Stata Journal* 10: 540–567.
- Xu, J., and J. S. Long. 2005. Confidence intervals for predicted outcomes in regression models for categorical outcomes. *Stata Journal* 5: 537–559.

Also see

- [R] **oprobit postestimation** — Postestimation tools for `oprobit`
- [R] **heckoprobit** — Ordered probit model with sample selection
- [R] **hetoprobit** — Heteroskedastic ordered probit regression
- [R] **logistic** — Logistic regression, reporting odds ratios
- [R] **mlogit** — Multinomial (polytomous) logistic regression
- [R] **mprobit** — Multinomial probit regression
- [R] **ologit** — Ordered logistic regression
- [R] **probit** — Probit regression
- [R] **zioprobit** — Zero-inflated ordered probit regression
- [BAYES] **bayes: oprobit** — Bayesian ordered probit regression
- [CM] **cmoprobit** — Rank-ordered probit choice model
- [ERM] **eoprobit** — Extended ordered probit regression
- [FMM] **fmm: oprobit** — Finite mixtures of ordered probit regression models
- [ME] **meoprobit** — Multilevel mixed-effects ordered probit regression
- [MI] **Estimation** — Estimation commands for use with `mi` estimate
- [SVY] **svy estimation** — Estimation commands for survey data
- [XT] **xtoprobit** — Random-effects ordered probit models
- [U] **20 Estimation and postestimation commands**

