oprobit — Ordered probit regression

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

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  depvar  [  indepvars  ]  [  if  ]  [  in  ]  [  weight  ]  [  ,  options  ]

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>offset(varname)  include varname in model with coefficient constrained to 1</td>
</tr>
<tr>
<td></td>
<td>constraints(constraints)  apply specified linear constraints</td>
</tr>
<tr>
<td>SE/Robust</td>
<td>vce(vcetype)  vcetype may be oim, robust, cluster clustvar, bootstrap, or</td>
</tr>
<tr>
<td></td>
<td>jackknife</td>
</tr>
<tr>
<td>Reporting</td>
<td>level(#)  set confidence level; default is level(95)</td>
</tr>
<tr>
<td></td>
<td>nocnsreport  do not display constraints</td>
</tr>
<tr>
<td></td>
<td>display_options  control columns and column formats, row spacing, line width,</td>
</tr>
<tr>
<td></td>
<td>display of omitted variables and base and empty cells, and</td>
</tr>
<tr>
<td></td>
<td>factor-variable labeling</td>
</tr>
<tr>
<td>Maximization</td>
<td>maximize_options  control the maximization process; seldom used</td>
</tr>
<tr>
<td></td>
<td>collinear  keep collinear variables</td>
</tr>
<tr>
<td></td>
<td>coeflegend  display legend instead of statistics</td>
</tr>
</tbody>
</table>

indepvars may contain factor variables; see [U] 11.4.3 Factor variables.

depvar and indepvars may contain time-series operators; see [U] 11.4.4 Time-series varlists.
bayes, bootstrap, by, fmm, fp, jackknife, mfp, mi estimate, nestreg, rolling, statsby, stepwise,
and svy are allowed; see [U] 11.1.10 Prefix commands. For more details, see [BAYES] bayes: oprobit and
[FMM] fmm: oprobit.

vce(bootstrap) and vce(jackknife) are not allowed with the mi estimate prefix; see [MI] mi estimate.
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; see [U] 11.1.6 weight.
collinear and coeflegend do not appear in the dialog box.

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

Options

<table>
<thead>
<tr>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>offset(varname), constraints(constraints); see [R] Estimation options.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SE/Robust</th>
</tr>
</thead>
<tbody>
<tr>
<td>vce(vcetype)  specifies the type of standard error reported, which includes types that are derived</td>
</tr>
<tr>
<td>from asymptotic theory (oim), that are robust to some kinds of misspecification (robust), that</td>
</tr>
<tr>
<td>allow for intragroup correlation (cluster clustvar), and that use bootstrap or jackknife methods</td>
</tr>
<tr>
<td>(bootstrap, jackknife); see [R] vce_option.</td>
</tr>
</tbody>
</table>
Reporting

level(#); see [R] Estimation options.

 nocnsreport; see [R] Estimation options.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nolabel, 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.

The following options are available with oprobit but is not shown in the dialog box:
collinear, coeflegend; see [R] Estimation options.

Remarks and examples

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 \( \beta_1, \beta_2, \ldots, \beta_k \) together with the cutpoints \( \kappa_1, \kappa_2, \ldots, \kappa_{I-1} \), where \( I \) is the number of possible outcomes. \( \kappa_0 \) is taken as \( -\infty \), and \( \kappa_I \) 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/r16/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

Ordered probit regression  Number of obs =  66
LR chi2(3) =  23.75
Prob > chi2 =  0.0000
Log likelihood = -78.020025  Pseudo R2 =  0.1321

rep77            Coef.   Std. Err.     z    P>|z|     [95% Conf. Interval]
-------------      --------      ------    ------    ----------------------
 foreign         1.704861     .4246796    4.01    0.000     0.8725037    2.537217
 length          0.0468675   .012648    3.71    0.000     0.022078     0.0716571
 mpg             0.1304559   .0378628    3.45    0.001     0.0562463    0.2046656
 /cut1           10.1589     3.076754    4.12     0.000    16.18923     16.18923
 /cut2           11.21003    3.107527    5.12     0.000    17.30067     17.30067
 /cut3           12.54561    3.155233    6.36     0.000    18.72975     18.72975
 /cut4           13.98059    3.218793    7.67     0.000    20.28931     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

- e(N)  number of observations
- e(N_cd)  number of completely determined observations
- e(k_cat)  number of categories
- e(k)  number of parameters
- e(k_aux)  number of auxiliary parameters
- e(k_eq)  number of equations in e(b)
- e(k_eq_model)  number of equations in overall model test
- e(k_dv)  number of dependent variables
- e(df_m)  model degrees of freedom
- e(r2_p)  pseudo-R-squared
- e(ll)  log likelihood
- e(ll_0)  log likelihood, constant-only model
- e(N_clust)  number of clusters
- e(chi2)  \( \chi^2 \)
- e(p)  p-value for model test
- e(rank)  rank of e(V)
- e(ic)  number of iterations
- e(rc)  return code
- e(converged)  1 if converged, 0 otherwise

Macros

- e(cmd)  oprobit
- e(cmdline)  command as typed
- e(depvar)  name of dependent variable
- e(vtype)  weight type
Methods and formulas

See Methods and formulas of [R] ologit.

References


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] 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] cmproprobit — 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