biprobit — Bivariate probit regression

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

Description

`biprobit` fits maximum-likelihood two-equation probit models—either a bivariate probit or a seemingly unrelated probit (limited to two equations).

Quick start

Bivariate probit regression of `y1` and `y2` on `x1`

```
biprobit y1 y2 x1
```

Bivariate probit regression of `y1` and `y2` on `x1`, `x2`, and `x3`

```
biprobit y1 y2 x1 x2 x3
```

Constrain the coefficients for `x1` to equality in both equations

```
constraint define 1 _b[y1:x1] = _b[y2:x1]
biprobit y1 y2 x1 x2 x3, constraints(1)
```

Seemingly unrelated bivariate probit regression

```
biprobit (y1 = x1 x2 x3) (y2 = x1 x2)
```

With robust standard errors

```
biprobit (y1 = x1 x2 x3) (y2 = x1 x2), vce(robust)
```

Poirier partial observability model with `difficult` option

```
biprobit (y1 = x1 x2) (y2 = x2 x3), partial difficult
```

Menu

biprobit

Statistics > Binary outcomes > Bivariate probit regression

Seemingly unrelated biprobit

Statistics > Binary outcomes > Seemingly unrelated bivariate probit regression

Syntax

Bivariate probit regression

`biprobit` *depvar*$_1$ *depvar*$_2$ [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

Seemingly unrelated bivariate probit regression

`biprobit` *equation*$_1$ *equation*$_2$ [*if*] [*in*] [*weight*] [, *su_options*]

where *equation*$_1$ and *equation*$_2$ are specified as

`(` [*eqname*`:`] *depvar* [=] [*indepvars*] [, `noconstant` `offset(`*varname*`)`] `)`

options	Description
Model	
`noconstant`	suppress constant term
`partial`	fit partial observability model
`offset1(`*varname*`)`	offset variable for first equation
`offset2(`*varname*`)`	offset variable for second equation
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `opg`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

su_options	Description
Model	
partial	fit partial observability model
constraints(*constraints*)	apply specified linear constraints
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, opg, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
lrmodel	perform the likelihood-ratio model test instead of the default Wald test
nocnsreport	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
collinear	keep collinear variables
coeflegend	display legend instead of statistics

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

$depvar_1$, $depvar_2$, *indepvars*, and *depvar* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

bayes, bayesboot, bootstrap, by, collect, fp, jackknife, rolling, statsby, and svy are allowed; see [U] **11.1.10 Prefix commands**. For more details, see [BAYES] **bayes: biprobit**.

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

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

pweights, fweights, and iweights 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

Model

noconstant; see [R] **Estimation options**.

partial specifies that the partial observability model be fit. This particular model commonly has poor convergence properties, so we recommend that you use the difficult option if you want to fit the Poirier partial observability model; see [R] **Maximize**.

This model computes the product of the two dependent variables so that you do not have to replace each with the product.

offset1(*varname*), offset2(*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(#), lrmodel, 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(#), nonrtolerance, and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

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

The following options are available with biprobit but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

For a good introduction to the bivariate probit models, see Greene (2018, sec. 17.9) and Pindyck and Rubinfeld (1998). Poirier (1980) explains the partial observability model. Van de Ven and Van Pragg (1981) explain the probit model with sample selection; see [R] **heckprobit** for details.

▷ Example 1

We use the data from Pindyck and Rubinfeld (1998, 332). In this dataset, the variables are whether children attend private school (private), number of years the family has been at the present residence (years), log of property tax (logptax), log of income (loginc), and whether the head of the household voted for an increase in property taxes (vote).

We wish to model the bivariate outcomes of whether children attend private school and whether the head of the household voted for an increase in property tax based on the other covariates.

```
. use https://www.stata-press.com/data/r19/school
. biprobit private vote years logptax loginc
Fitting comparison equation 1:
Iteration 0:  Log likelihood = -31.967097
Iteration 1:  Log likelihood = -31.452424
Iteration 2:  Log likelihood = -31.448958
Iteration 3:  Log likelihood = -31.448958
Fitting comparison equation 2:
Iteration 0:  Log likelihood = -63.036914
Iteration 1:  Log likelihood = -58.534843
Iteration 2:  Log likelihood = -58.497292
Iteration 3:  Log likelihood = -58.497288
Comparison:   Log likelihood = -89.946246
Fitting full model:
Iteration 0:  Log likelihood = -89.946246
Iteration 1:  Log likelihood = -89.258897
Iteration 2:  Log likelihood = -89.254028
Iteration 3:  Log likelihood = -89.254028
Bivariate probit regression                         Number of obs =     95
                                                    Wald chi2(6)  =   9.59
Log likelihood = -89.254028                         Prob > chi2   = 0.1431

------------------------------------------------------------------------------
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
private      |
       years |  -.0118884   .0256778    -0.46   0.643    -.0622159    .0384391
     logptax |  -.1066962   .6669782    -0.16   0.873    -1.413949    1.200557
      loginc |   .3762037   .5306484     0.71   0.478     -.663848    1.416255
       _cons |  -4.184694   4.837817    -0.86   0.387    -13.66664    5.297253
-------------+----------------------------------------------------------------
vote         |
       years |  -.0168561   .0147834    -1.14   0.254    -.0458309    .0121188
     logptax |  -1.288707   .5752266    -2.24   0.025    -2.416131   -.1612839
      loginc |    .998286   .4403565     2.27   0.023     .1352031    1.861369
       _cons |  -.5360573   4.068509    -0.13   0.895    -8.510188    7.438073
-------------+----------------------------------------------------------------
     /athrho |  -.2764525   .2412099    -1.15   0.252    -.7492153    .1963102
-------------+----------------------------------------------------------------
         rho |  -.2696186   .2236753                     -.6346806    .1938267
------------------------------------------------------------------------------
LR test of rho=0: chi2(1) = 1.38444                       Prob > chi2 = 0.2393
```

The output shows several iteration logs. The first iteration log corresponds to running the univariate probit model for the first equation, and the second log corresponds to running the univariate probit for the second model. If $\rho = 0$, the sum of the log likelihoods from these two models will equal the log likelihood of the bivariate probit model; this sum is printed in the iteration log as the comparison log likelihood.

The final iteration log is for fitting the full bivariate probit model. A likelihood-ratio test of the log likelihood for this model and the comparison log likelihood is presented at the end of the output. If we had specified the `vce(robust)` option, this test would be presented as a Wald test instead of as a likelihood-ratio test.

We could have fit the same model by using the seemingly unrelated syntax as

```
. biprobit (private=years logptax loginc) (vote=years logptax loginc)
```

◁

Stored results

biprobit stores the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_aux)	number of auxiliary parameters
e(k_eq_model)	number of equations in overall model test
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model (lrmodel only)
e(ll_c)	log likelihood, comparison model
e(N_clust)	number of clusters
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(p)	p-value for model test
e(rho)	ρ
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	biprobit
e(cmdline)	command as typed
e(depvar)	names of dependent variables
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(offset1)	offset for first equation
e(offset2)	offset for second equation
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
d(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

`e(sample)`	marks estimation sample

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

Matrices

`r(table)`	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 log likelihood, $\ln L$, is given by

$$\begin{aligned}
\xi_j^\beta &= x_j\beta + \text{offset}_j^\beta \\
\xi_j^\gamma &= z_j\gamma + \text{offset}_j^\gamma \\
q_{1j} &= \begin{cases} 1 & \text{if } y_{1j} \neq 0 \\ -1 & otherwise \end{cases} \\
q_{2j} &= \begin{cases} 1 & \text{if } y_{2j} \neq 0 \\ -1 & otherwise \end{cases} \\
\rho_j^* &= q_{1j}q_{2j}\rho \\
\ln L &= \sum_{j=1}^{n} w_j \ln\Phi_2\left(q_{1j}\xi_j^\beta, q_{2j}\xi_j^\gamma, \rho_j^*\right)
\end{aligned}$$

where $\Phi_2(\cdot)$ is the cumulative bivariate normal distribution function (with mean $[0\ \ 0]'$) and w_j is an optional weight for observation j. This derivation assumes that

$$\begin{aligned}
y_{1j}^* &= x_j\beta + \epsilon_{1j} + \text{offset}_j^\beta \\
y_{2j}^* &= z_j\gamma + \epsilon_{2j} + \text{offset}_j^\gamma \\
E(\epsilon_1) &= E(\epsilon_2) = 0 \\
\text{Var}(\epsilon_1) &= \text{Var}(\epsilon_2) = 1 \\
\text{Cov}(\epsilon_1, \epsilon_2) &= \rho
\end{aligned}$$

where y_{1j}^* and y_{2j}^* are the unobserved latent variables; instead, we observe only $y_{ij} = 1$ if $y_{ij}^* > 0$ and $y_{ij} = 0$ otherwise (for $i = 1, 2$).

In the maximum likelihood estimation, ρ is not directly estimated, but atanh ρ is

$$\text{atanh}\ \rho = \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right)$$

From the form of the likelihood, if $\rho = 0$, then the log likelihood for the bivariate probit models is equal to the sum of the log likelihoods of the two univariate probit models. A likelihood-ratio test may therefore be performed by comparing the likelihood of the full bivariate model with the sum of the log likelihoods for the univariate probit models.

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

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

References

De Luca, G. 2008. SNP and SML estimation of univariate and bivariate binary-choice models. *Stata Journal* 8: 190–220.

Greene, W. H. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Heckman, J. J. 1979. Sample selection bias as a specification error. *Econometrica* 47: 153–161. https://doi.org/10.2307/1912352.

Hernández-Alava, M., and S. Pudney. 2016. bicop: A command for fitting bivariate ordinal regressions with residual dependence characterized by a copula function and normal mixture marginals. *Stata Journal* 16: 159–184.

Lokshin, M., and Z. Sajaia. 2011. Impact of interventions on discrete outcomes: Maximum likelihood estimation of the binary choice models with binary endogenous regressors. *Stata Journal* 11: 368–385.

Mullahy, J. 2016. Estimation of multivariate probit models via bivariate probit. *Stata Journal* 16: 37–51.

Pindyck, R. S., and D. L. Rubinfeld. 1998. *Econometric Models and Economic Forecasts*. 4th ed. New York: McGraw–Hill.

Poirier, D. J. 1980. Partial observability in bivariate probit models. *Journal of Econometrics* 12: 209–217. https://doi.org/10.1016/0304-4076(80)90007-X.

Van de Ven, W. P. M. M., and B. M. S. Van Pragg. 1981. The demand for deductibles in private health insurance: A probit model with sample selection. *Journal of Econometrics* 17: 229–252. https://doi.org/10.1016/0304-4076(81)90028-2.

Also see

[R] **biprobit postestimation** — Postestimation tools for biprobit

[R] **mprobit** — Multinomial probit regression

[R] **probit** — Probit regression

[BAYES] **bayes: biprobit** — Bayesian bivariate probit regression

[SVY] **svy estimation** — Estimation commands for survey data

[U] 20 Estimation and postestimation commands

For suggested citations, see the FAQ on citing Stata documentation.