Title

heckprobit — Probit model with sample selection

Description

`heckprobit` fits maximum-likelihood probit models with sample selection.

Quick start

Probit model of `y` on `x` with sample selection indicated by binary variable `selected` and predicted by `v`

```
heckprobit y x, select(selected = v x)
```

Suppress iteration log

```
heckprobit y x, select(selected = v x) nolog
```

With cluster–robust standard errors for clustering by levels of `cvar`

```
heckprobit y x, select(selected = v x) vce(cluster cvar)
```

Menu

Statistics > Sample-selection models > Probit model with sample selection

Syntax

`heckprobit` *depvar indepvars* [*if*] [*in*] [*weight*] ,

    `select(` [*depvar*$_s$ =] *varlist*$_s$ [, `noconstant` `offset(`*varname*$_o$`)`] `)` [*options*]

options	Description
Model	
* `select()`	specify selection equation: dependent and independent variables; whether to have constant term and offset variable
`noconstant`	suppress constant term
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`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)`
`first`	report first-step probit estimates
`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

*`select()` is required.

  The full specification is `select(` [*depvar*$_s$ =] *varlist*$_s$ [, `noconstant` `offset(`*varname*$_o$`)`] `)`.

indepvars and *varlist*$_s$ may contain factor variables; see **[U] 11.4.3 Factor variables**.

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

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

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

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

`pweight`s, `fweight`s, and `iweight`s 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

`select(` [*depvar*$_s$ =] *varlist*$_s$ [, `noconstant offset(`*varname*$_o$`)`] `)` specifies the variables and options for the selection equation. It is an integral part of specifying a selection model and is required. The selection equation should contain at least one variable that is not in the outcome equation.

If *depvar*$_s$ is specified, it should be coded as 0 or 1, 0 indicating an observation not selected and 1 indicating a selected observation. If *depvar*$_s$ is not specified, observations for which *depvar* is not missing are assumed selected, and those for which *depvar* is missing are assumed not selected.

`noconstant` suppresses the selection constant term (intercept).

`offset(`*varname*$_o$`)` specifies that selection offset *varname*$_o$ be included in the model with the coefficient constrained to be 1.

`noconstant`, `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**.

`first` specifies that the first-step probit estimates of the selection equation be displayed before estimation.

`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 `heckprobit` but are not shown in the dialog box:

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

Remarks and examples

stata.com

The probit model with sample selection (Van de Ven and Van Pragg 1981) assumes that there exists an underlying relationship

$$y_j^* = \mathbf{x}_j\boldsymbol{\beta} + u_{1j} \qquad \textit{latent equation}$$

such that we observe only the binary outcome

$$y_j^{\text{probit}} = (y_j^* > 0) \qquad \textit{probit equation}$$

The dependent variable, however, is not always observed. Rather, the dependent variable for observation j is observed if

$$y_j^{\text{select}} = (\mathbf{z}_j\boldsymbol{\gamma} + u_{2j} > 0) \qquad \textit{selection equation}$$

where

$$u_1 \sim N(0,1)$$

$$u_2 \sim N(0,1)$$

$$\text{corr}(u_1, u_2) = \rho$$

When $\rho \neq 0$, standard probit techniques applied to the first equation yield biased results. `heckprobit` provides consistent, asymptotically efficient estimates for all the parameters in such models.

For the model to be well identified, the selection equation should have at least one variable that is not in the probit equation. Otherwise, the model is identified only by functional form, and the coefficients have no structural interpretation.

▷ Example 1

We use the data from Pindyck and Rubinfeld (1998). 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 one voted for an increase in property taxes (`vote`).

In this example, we alter the meaning of the data. Here we assume that we observe whether children attend private school only if the family votes for increasing the property taxes. This assumption is not true in the dataset, and we make it only to illustrate the use of this command.

We observe whether children attend private school only if the head of household voted for an increase in property taxes. We assume that the vote is affected by the number of years in residence, the current property taxes paid, and the household income. We wish to model whether children are sent to private school on the basis of the number of years spent in the current residence and the current property taxes paid.

```
. use https://www.stata-press.com/data/r18/school
. heckprobit private years logptax, select(vote=years loginc logptax)
Fitting probit model:
Iteration 0:  Log likelihood = -17.122381
Iteration 1:  Log likelihood = -16.243974
  (output omitted)
Iteration 5:  Log likelihood = -15.883655
Fitting selection model:
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 = -74.380943
Fitting starting values:
Iteration 0:  Log likelihood = -40.895684
Iteration 1:  Log likelihood = -16.654497
  (output omitted)
Iteration 6:  Log likelihood = -15.753765
Fitting full model:
Iteration 0:  Log likelihood = -75.010619  (not concave)
Iteration 1:  Log likelihood = -74.287758
Iteration 2:  Log likelihood = -74.250143
Iteration 3:  Log likelihood = -74.245088
Iteration 4:  Log likelihood = -74.244973
Iteration 5:  Log likelihood = -74.244973
Probit model with sample selection              Number of obs     =         95
                                                      Selected    =         59
                                                      Nonselected =         36

                                                Wald chi2(2)      =       1.04
Log likelihood = -74.24497                      Prob > chi2       =     0.5935
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
private						
years	-.1142596	.1461715	-0.78	0.434	-.4007505	.1722313
logptax	.3516101	1.016483	0.35	0.729	-1.64066	2.34388
_cons	-2.780667	6.905827	-0.40	0.687	-16.31584	10.75451
vote						
years	-.0167511	.0147735	-1.13	0.257	-.0457067	.0122045
loginc	.9923023	.4430008	2.24	0.025	.1240368	1.860568
logptax	-1.278783	.5717545	-2.24	0.025	-2.399401	-.1581646
_cons	-.5458205	4.070417	-0.13	0.893	-8.523692	7.432051
/athrho	-.8663164	1.450017	-0.60	0.550	-3.708298	1.975665
rho	-.6994978	.7405281			-.9987983	.9622674

```
LR test of indep. eqns. (rho = 0): chi2(1) = 0.27         Prob > chi2 = 0.6020
```

The output shows several iteration logs. The first iteration log corresponds to running the probit model for those observations in the sample where we have observed the outcome. The second iteration log corresponds to running the selection probit model, which models whether we observe our outcome of interest. If $\rho = 0$, the sum of the log likelihoods from these two models will equal the log likelihood of the probit model with sample selection; this sum is printed in the iteration log as the comparison log likelihood. The third iteration log shows starting values for the iterations.

The final iteration log is for fitting the full probit model with sample selection. 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.

◁

▷ Example 2

In example 1, we could have obtained robust standard errors by specifying the `vce(robust)` option. We do this here and also eliminate the iteration logs by using the `nolog` option:

```
. heckprobit private years logptax, sel(vote=years loginc logptax) vce(robust)
> nolog
Probit model with sample selection                Number of obs     =         95
                                                        Selected    =         59
                                                        Nonselected =         36
                                                  Wald chi2(2)      =       2.55
Log pseudolikelihood = -74.24497                  Prob > chi2       =     0.2798
```

	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
private						
years	-.1142596	.1113968	-1.03	0.305	-.3325934	.1040741
logptax	.3516101	.7358211	0.48	0.633	-1.090573	1.793793
_cons	-2.780667	4.786652	-0.58	0.561	-12.16233	6.600998
vote						
years	-.0167511	.0173344	-0.97	0.334	-.0507259	.0172237
loginc	.9923023	.4228042	2.35	0.019	.1636213	1.820983
logptax	-1.278783	.5095156	-2.51	0.012	-2.277415	-.2801506
_cons	-.5458205	4.54389	-0.12	0.904	-9.451682	8.360041
/athrho	-.8663164	1.63062	-0.53	0.595	-4.062272	2.329639
rho	-.6994978	.8327621			-.9994078	.9812312

```
Wald test of indep. eqns. (rho = 0): chi2(1) = 0.28        Prob > chi2 = 0.5952
```

Regardless of whether we specify the `vce(robust)` option, the outcome is not significantly different from the outcome obtained by fitting the probit and selection models separately. This result is not surprising because the selection mechanism estimated was invented for the example rather than borne from any economic theory.

◁

Stored results

heckprobit stores the following in e():

Scalars	
e(N)	number of observations
e(N_selected)	number of selected observations
e(N_nonselected)	number of nonselected observations
e(k)	number of parameters
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_aux)	number of auxiliary parameters
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
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(p_c)	p-value for comparison 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)	heckprobit
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 regression equation
e(offset2)	offset for selection equation
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	type of comparison χ^2 test
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
e(predict)	program used to implement predict
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

Van de Ven and Van Pragg (1981) provide an introduction and an explanation of this model.

The probit equation is

$$y_j = (\mathbf{x}_j\boldsymbol{\beta} + u_{1j} > 0)$$

The selection equation is

$$\mathbf{z}_j\boldsymbol{\gamma} + u_{2j} > 0$$

where

$$u_1 \sim N(0,1)$$
$$u_2 \sim N(0,1)$$
$$\mathrm{corr}(u_1, u_2) = \rho$$

The log likelihood is

$$\begin{aligned} \ln L = & \sum_{\substack{j\in S \\ y_j\neq 0}} w_j \ln\left\{\Phi_2\left(x_j\beta + \mathrm{offset}_j^{\beta}, z_j\gamma + \mathrm{offset}_j^{\gamma}, \rho\right)\right\} \\ & + \sum_{\substack{j\in S \\ y_j=0}} w_j \ln\left\{\Phi_2\left(-x_j\beta + \mathrm{offset}_j^{\beta}, z_j\gamma + \mathrm{offset}_j^{\gamma}, -\rho\right)\right\} \\ & + \sum_{j\notin S} w_j \ln\left\{1 - \Phi\left(z_j\gamma + \mathrm{offset}_j^{\gamma}\right)\right\} \end{aligned}$$

where S is the set of observations for which y_j is observed, $\Phi_2(\cdot)$ is the cumulative bivariate normal distribution function (with mean $[\,0\ \ 0\,]'$), $\Phi(\cdot)$ is the standard cumulative normal, and w_j is an optional weight for observation j.

In the maximum likelihood estimation, ρ is not directly estimated. Directly estimated is atanh ρ:

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

From the form of the likelihood, it is clear that if $\rho = 0$, the log likelihood for the probit model with sample selection is equal to the sum of the probit model for the outcome y and the selection model. We can perform a likelihood-ratio test by comparing the likelihood of the full model with the sum of the log likelihoods for the probit and selection 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*.

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

References

Baum, C. F. 2006. *An Introduction to Modern Econometrics Using Stata*. College Station, TX: Stata Press.

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

De Luca, G., and V. Perotti. 2011. Estimation of ordered response models with sample selection. *Stata Journal* 11: 213–239.

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

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.

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.

Muro, J., C. Suárez, and M. Zamora. 2010. Computing Murphy–Topel-corrected variances in a heckprobit model with endogeneity. *Stata Journal* 10: 252–258.

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

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] **heckprobit postestimation** — Postestimation tools for heckprobit

[R] **heckman** — Heckman selection model

[R] **heckoprobit** — Ordered probit model with sample selection

[R] **heckpoisson** — Poisson regression with sample selection

[R] **probit** — Probit regression

[BAYES] **bayes: heckprobit** — Bayesian probit model with sample selection

[CAUSAL] **etregress** — Linear regression with endogenous treatment effects

[ERM] **eprobit** — Extended 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.