probit — Probit regression

Description

`probit` fits a probit model for a binary dependent variable, assuming that the probability of a positive outcome is determined by the standard normal cumulative distribution function. `probit` can compute robust and cluster–robust standard errors and adjust results for complex survey designs.

Quick start

Probit model of `y` on continuous variable `x1`

```
probit y x1
```

Add square of `x1`

```
probit y c.x1##c.x1
```

Same as above, but report bootstrap standard errors

```
probit y c.x1##c.x1, vce(bootstrap)
```

Bootstrap estimates of coefficients

```
bootstrap _b: probit y c.x1##c.x1
```

Adjust for complex survey design using `svyset` data and add `x2`

```
svy: probit y c.x1##c.x1 x2
```

Menu

Statistics > Binary outcomes > Probit regression

Syntax

`probit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`noconstant`	suppress constant term
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`asis`	retain perfect predictor variables
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `opg`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`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
`nocoef`	do not display the coefficient table; 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 and *indepvars* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

`bayes`, `bayesboot`, `bootstrap`, `by`, `collect`, `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: probit** and [FMM] **fmm: probit**.

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

`fweight`s, `iweight`s, and `pweight`s are allowed; see [U] **11.1.6 weight**.

`nocoef`, `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, offset(*varname*), constraints(*constraints*); see [R] **Estimation options**.

asis specifies that all specified variables and observations be retained in the maximization process. This option is typically not specified and may introduce numerical instability. Normally probit omits variables that perfectly predict success or failure in the dependent variable along with their associated observations. In those cases, the effective coefficient on the omitted variables is infinity (negative infinity) for variables that completely determine a success (failure). Dropping the variable and perfectly predicted observations has no effect on the likelihood or estimates of the remaining coefficients and increases the numerical stability of the optimization process. Specifying this option forces retention of perfect predictor variables and their associated observations.

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

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

nocoef specifies that the coefficient table not be displayed. This option is sometimes used by programmers but is of no use interactively.

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

Remarks and examples

Remarks are presented under the following headings:

probit fits maximum likelihood models with dichotomous dependent (left-hand-side) variables coded as 0/1 (more precisely, coded as 0 and not 0).

For grouped data or data in binomial form, a probit model can be fit using glm with the family(binomial) and link(probit) options.

▷ Example 1

We have data on the make, weight, and mileage rating of 22 foreign and 52 domestic automobiles. We wish to fit a probit model explaining whether a car is foreign based on its weight and mileage. Here is an overview of our data:

```
. use https://www.stata-press.com/data/r19/auto
(1978 automobile data)

. keep make mpg weight foreign

. describe

Contains data from https://www.stata-press.com/data/r19/auto.dta
 Observations:            74                  1978 automobile data
    Variables:             4                  13 Apr 2024 17:45
                                              (_dta has notes)
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
make            str18   %-18s                 Make and model
mpg             int     %8.0g                 Mileage (mpg)
weight          int     %8.0gc                Weight (lbs.)
foreign         byte    %8.0g      origin     Car origin
-------------------------------------------------------------------------------
Sorted by: foreign
     Note: Dataset has changed since last saved.

. inspect foreign

foreign:  Car origin                          Number of observations
--------------------                   ---------------------------------------
                                            Total      Integers   Nonintegers
|  #                          Negative          -             -             -
|  #                          Zero             52            52             -
|  #                          Positive         22            22             -
|  #                                   ----------   -----------   -----------
|  #   #                      Total            74            74             -
|  #   #                      Missing           -
+-----------------------               ----------
0                     1                        74
   (2 unique values)

      foreign is labeled and all values are documented in the label.
```

The foreign variable takes on two unique values, 0 and 1. The value 0 denotes a domestic car, and 1 denotes a foreign car.

The model that we wish to fit is

$$\Pr(\texttt{foreign} = 1) = \Phi(\beta_0 + \beta_1\texttt{weight} + \beta_2\texttt{mpg})$$

where Φ is the cumulative normal distribution.

To fit this model, we type

```
. probit foreign weight mpg
Iteration 0:  Log likelihood =  -45.03321
Iteration 1:  Log likelihood = -27.914626
  (output omitted)
Iteration 5:  Log likelihood = -26.844189
Probit regression                                       Number of obs =     74
                                                        LR chi2(2)    =  36.38
                                                        Prob > chi2   = 0.0000
Log likelihood = -26.844189                             Pseudo R2     = 0.4039

------------------------------------------------------------------------------
     foreign | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      weight |  -.0023355   .0005661    -4.13   0.000     -.003445   -.0012261
         mpg |  -.1039503   .0515689    -2.02   0.044    -.2050235   -.0028772
       _cons |   8.275464   2.554142     3.24   0.001     3.269437    13.28149
------------------------------------------------------------------------------
```

We find that heavier cars are less likely to be foreign and that cars yielding better gas mileage are also less likely to be foreign, at least holding the weight of the car constant.

See [R] **Maximize** for an explanation of the output.

◁

❑ Technical note

Stata interprets a value of 0 as a negative outcome (failure) and treats all other values (except missing) as positive outcomes (successes). Thus if your dependent variable takes on the values 0 and 1, then 0 is interpreted as failure and 1 as success. If your dependent variable takes on the values 0, 1, and 2, then 0 is still interpreted as failure, but both 1 and 2 are treated as successes.

If you prefer a more formal mathematical statement, when you type `probit` y x, Stata fits the model

$$\Pr(y_j \neq 0 \mid \mathbf{x}_j) = \Phi(\mathbf{x}_j\boldsymbol{\beta})$$

where Φ is the standard cumulative normal.

❑

Robust standard errors

If you specify the vce(robust) option, probit reports robust standard errors; see [U] 20.22 Obtaining robust variance estimates.

▷ Example 2

For the model from example 1, the robust calculation increases the standard error of the coefficient on mpg by almost 15%:

```
. probit foreign weight mpg, vce(robust) nolog
Probit regression                                       Number of obs =     74
                                                        Wald chi2(2)  =  30.26
                                                        Prob > chi2   = 0.0000
Log pseudolikelihood = -26.844189                       Pseudo R2     = 0.4039

------------------------------------------------------------------------------
             |               Robust
     foreign | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      weight |  -.0023355   .0004934    -4.73   0.000    -.0033025   -.0013686
         mpg |  -.1039503   .0593548    -1.75   0.080    -.2202836    .0123829
       _cons |   8.275464   2.539177     3.26   0.001     3.298769    13.25216
------------------------------------------------------------------------------
```

Without vce(robust), the standard error for the coefficient on mpg was reported to be 0.052 with a resulting confidence interval of $[-0.21, -0.00]$.

◁

▷ Example 3

The `vce(cluster` *clustvar*`)` option can relax the independence assumption required by the probit estimator to independence between clusters. To demonstrate, we will switch to a different dataset.

We are studying unionization of women in the United States and have a dataset with 26,200 observations on 4,434 women between 1970 and 1988. We will use the variables `age` (the women were 14–26 in 1968, and our data span the age range of 16–46), `grade` (years of schooling completed, ranging from 0 to 18), `not_smsa` (28% of the person-time was spent living outside an SMSA—standard metropolitan statistical area), `south` (41% of the person-time was in the South), and `year`. Each of these variables is included in the regression as a covariate along with the interaction between `south` and `year`. This interaction, along with the `south` and `year` variables, is specified in the `probit` command using factor-variables notation, `south##c.year`. We also have variable `union`, indicating union membership. Overall, 22% of the person-time is marked as time under union membership, and 44% of these women have belonged to a union.

We fit the following model, ignoring that women are observed an average of 5.9 times each in these data:

```
. use https://www.stata-press.com/data/r19/union, clear
(NLS Women 14-24 in 1968)

. probit union age grade not_smsa south##c.year
Iteration 0:  Log likelihood =  -13864.23
Iteration 1:  Log likelihood = -13545.541
Iteration 2:  Log likelihood = -13544.385
Iteration 3:  Log likelihood = -13544.385

Probit regression                                       Number of obs = 26,200
                                                        LR chi2(6)    = 639.69
                                                        Prob > chi2   = 0.0000
Log likelihood = -13544.385                             Pseudo R2     = 0.0231
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
age	.0118481	.0029072	4.08	0.000	.0061502	.017546
grade	.0267365	.0036689	7.29	0.000	.0195457	.0339273
not_smsa	-.1293525	.0202595	-6.38	0.000	-.1690604	-.0896445
1.south	-.8281077	.2472219	-3.35	0.001	-1.312654	-.3435618
year	-.0080931	.0033469	-2.42	0.016	-.0146529	-.0015333
south#c.year						
1	.0057369	.0030917	1.86	0.064	-.0003226	.0117965
_cons	-.6542487	.2007777	-3.26	0.001	-1.047766	-.2607316

The reported standard errors in this model are probably meaningless. Women are observed repeatedly, and so the observations are not independent. Looking at the coefficients, we find a large southern effect against unionization and a time trend for the south that is almost significantly different from the overall downward trend. The `vce(cluster` *clustvar*`)` option provides a way to fit this model and obtains correct standard errors:

```
. probit union age grade not_smsa south##c.year, vce(cluster id)
Iteration 0:  Log pseudolikelihood =  -13864.23
Iteration 1:  Log pseudolikelihood = -13545.541
Iteration 2:  Log pseudolikelihood = -13544.385
Iteration 3:  Log pseudolikelihood = -13544.385
Probit regression                                       Number of obs = 26,200
                                                        Wald chi2(6)  = 166.53
                                                        Prob > chi2   = 0.0000
Log pseudolikelihood = -13544.385                       Pseudo R2     = 0.0231
                            (Std. err. adjusted for 4,434 clusters in idcode)
```

union	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
age	.0118481	.0056625	2.09	0.036	.0007499	.0229463
grade	.0267365	.0078124	3.42	0.001	.0114244	.0420486
not_smsa	-.1293525	.0403885	-3.20	0.001	-.2085125	-.0501925
1.south	-.8281077	.3201584	-2.59	0.010	-1.455607	-.2006089
year	-.0080931	.0060829	-1.33	0.183	-.0200153	.0038292
south#c.year						
1	.0057369	.0040133	1.43	0.153	-.002129	.0136029
_cons	-.6542487	.3485976	-1.88	0.061	-1.337487	.02899

These standard errors are larger than those reported by the inappropriate conventional calculation. By comparison, another model we could fit is an equal-correlation population-averaged probit model:

```
. xtprobit union age grade not_smsa south##c.year, pa
Iteration 1:  Tolerance = .12544249
Iteration 2:  Tolerance = .0034686
Iteration 3:  Tolerance = .00017448
Iteration 4:  Tolerance = 8.382e-06
Iteration 5:  Tolerance = 3.997e-07
GEE population-averaged model                        Number of obs    = 26,200
Group variable: idcode                               Number of groups =  4,434
Family: Binomial                                     Obs per group:
Link:    Probit                                                   min =      1
Correlation: exchangeable                                         avg =    5.9
                                                                  max =     12
                                                     Wald chi2(6)     = 242.57
Scale parameter = 1                                  Prob > chi2      = 0.0000
```

union	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
age	.0089699	.0053208	1.69	0.092	-.0014586	.0193985
grade	.0333174	.0062352	5.34	0.000	.0210966	.0455382
not_smsa	-.0715717	.027543	-2.60	0.009	-.1255551	-.0175884
1.south	-1.017368	.207931	-4.89	0.000	-1.424905	-.6098308
year	-.0062708	.0055314	-1.13	0.257	-.0171122	.0045706
south#c.year						
1	.0086294	.00258	3.34	0.001	.0035727	.013686
_cons	-.8670997	.294771	-2.94	0.003	-1.44484	-.2893592

The coefficient estimates are similar, but these standard errors are smaller than those produced by `probit, vce(cluster` *clustvar*`)`, as we would expect. If the equal-correlation assumption is valid, the population-averaged probit estimator above should be more efficient.

Is the assumption valid? That is a difficult question to answer. The default population-averaged estimates correspond to an assumption of exchangeable correlation within person. It would not be unreasonable to assume an AR(1) correlation within person or to assume that the observations are correlated but that we do not wish to impose any structure. See [XT] **xtprobit** and [XT] **xtgee** for full details.

◁

`probit, vce(cluster` *clustvar*`)` is robust to assumptions about within-cluster correlation. That is, it inefficiently sums within cluster for the standard error calculation rather than attempting to exploit what might be assumed about the within-cluster correlation.

Model identification

The `probit` command has one more feature that is probably the most useful. It will automatically check the model for identification and, if the model is underidentified, omit whatever variables and observations are necessary for estimation to proceed.

▷ Example 4

Have you ever fit a probit model where one or more of your independent variables perfectly predicted one or the other outcome?

For instance, consider the following data:

Outcome y	Independent variable x
0	1
0	1
0	0
1	0

Say that we wish to predict the outcome on the basis of the independent variable. The outcome is always zero when the independent variable is one. In our data, $\Pr(y = 0 \mid x = 1) = 1$, which means that the probit coefficient on x must be minus infinity with a corresponding infinite standard error. At this point, you may suspect that we have a problem.

Unfortunately, not all such problems are so easily detected, especially if you have many independent variables in your model. If you have ever had such difficulties, then you have experienced one of the more unpleasant aspects of computer optimization. The computer has no idea that it is trying to solve for an infinite coefficient as it begins its iterative process. All it knows is that, at each step, making the coefficient a little bigger, or a little smaller, works wonders. It continues on its merry way until either 1) the whole thing comes crashing to the ground when a numerical overflow error occurs or 2) it reaches some predetermined cutoff that stops the process. Meanwhile, you have been waiting. And the estimates that you finally receive, if any, may be nothing more than numerical roundoff.

Stata watches for these sorts of problems, alerts you, fixes them, and then properly fits the model.

Let's return to our automobile data. Among the variables we have in the data is one called `repair` that takes on three values. A value of 1 indicates that the car has a poor repair record, 2 indicates an average record, and 3 indicates a better-than-average record. Here is a tabulation of our data:

```
. use https://www.stata-press.com/data/r19/repair
(1978 automobile data)

. tabulate foreign repair

           |              Repair
Car origin |         1          2          3 |     Total
-----------+---------------------------------+----------
  Domestic |        10         27          9 |        46 
   Foreign |         0          3          9 |        12 
-----------+---------------------------------+----------
     Total |        10         30         18 |        58 
```

All the cars with poor repair records (`repair` = 1) are domestic. If we were to attempt to predict `foreign` on the basis of the repair records, the predicted probability for the `repair` = 1 category would have to be zero. This in turn means that the probit coefficient must be minus infinity, and that would set most computer programs buzzing.

Let's try using Stata on this problem.

```
. probit foreign b3.repair
note: 1.repair != 0 predicts failure perfectly;
      1.repair omitted and 10 obs not used.
Iteration 0:  Log likelihood = -26.992087
Iteration 1:  Log likelihood = -22.276479
Iteration 2:  Log likelihood = -22.229184
Iteration 3:  Log likelihood = -22.229138
Iteration 4:  Log likelihood = -22.229138
Probit regression                                       Number of obs =     48
                                                        LR chi2(1)    =   9.53
                                                        Prob > chi2   = 0.0020
Log likelihood = -22.229138                             Pseudo R2     = 0.1765

------------------------------------------------------------------------------
     foreign | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      repair |
          1  |          0  (empty)
          2  |  -1.281552   .4297326    -2.98   0.003    -2.123812   -.4392911
             |
       _cons |   1.16e-16    .295409     0.00   1.000     -.578991     .578991
------------------------------------------------------------------------------
```

Remember that all the cars with poor repair records (`repair` = 1) are domestic, so the model cannot be fit, or at least it cannot be fit if we restrict ourselves to finite coefficients. Stata noted that fact "note: 1.repair != 0 predicts failure perfectly". This is Stata's mathematically precise way of saying what we said in English. When `repair` is 1, the car is domestic.

Stata then went on to say, "1.repair omitted and 10 obs not used". This is Stata eliminating the problem. First, `1.repair` had to be removed from the model because it would have an infinite coefficient. Then, the 10 observations that led to the problem had to be eliminated, as well, so as not to bias the remaining coefficients in the model. The 10 observations that are not used are the 10 domestic cars that have poor repair records.

Stata then fit what was left of the model, using the remaining observations. Because no observations remained for cars with poor repair records, Stata reports "(empty)" in the row for `repair` = 1.

◁

❑ Technical note

Stata is pretty smart about catching these problems. It will catch "one-way causation by a dummy variable", as we demonstrated above.

Stata also watches for "two-way causation", that is, a variable that perfectly determines the outcome, both successes and failures. Here Stata says that the variable "predicts outcome perfectly" and stops. Statistics dictate that no model can be fit.

Stata also checks your data for collinear variables; it will say "so-and-so omitted because of collinearity". No observations need to be eliminated here and model fitting will proceed without the offending variable.

It will also catch a subtle problem that can arise with continuous data. For instance, if we were estimating the chances of surviving the first year after an operation, and if we included in our model `age`, and if all the persons over 65 died within the year, Stata will say, "age $>$ 65 predicts failure perfectly". It will then inform us about how it resolves the issue and fit what can be fit of our model.

`probit` (and `logit`, `logistic`, and `ivprobit`) will also occasionally fail to converge and then display messages such as

```
Note: 4 failures and 0 successes completely determined.
```

The cause of this message and what to do if you see it are described in [R] **logit**.

❑

Video examples

Probit regression with categorical covariates

Probit regression with continuous covariates

Probit regression with categorical and continuous covariates

Stored results

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

Scalars

`e(N)`	number of observations
`e(N_cds)`	number of completely determined successes
`e(N_cdf)`	number of completely determined failures
`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_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(r2_p)`	pseudo-R^2
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(N_clust)`	number of clusters
`e(chi2)`	χ^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)`	`probit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model 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(estat_cmd)`	program used to implement `estat`
`e(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(mns)`	vector of means of the independent variables
`e(rules)`	information about perfect predictors
`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

Probit analysis originated in connection with bioassay, and the word probit, a contraction of "probability unit", was suggested by Bliss (1934a, 1934b). For an introduction to probit and logit, see, for example, Aldrich and Nelson (1984), Cameron and Trivedi (2022), Long (1997), Pampel (2021), or Powers and Xie (2008). Long and Freese (2014, chap. 5 and 6) and Jones (2007, chap. 3) provide introductions to probit and logit, along with Stata examples.

The log-likelihood function for probit is

$$\ln L = \sum_{j \in S} w_j \ln\Phi(\mathbf{x}_j\boldsymbol{\beta}) + \sum_{j \notin S} w_j \ln\Big\{1 - \Phi(\mathbf{x}_j\boldsymbol{\beta})\Big\}$$

where Φ is the cumulative normal and w_j denotes the optional weights. $\ln L$ is maximized, as described in [R] **Maximize**.

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*. The scores are calculated as $\mathbf{u}_j = \{\phi(\mathbf{x}_j\mathbf{b})/\Phi(\mathbf{x}_j\mathbf{b})\}\mathbf{x}_j$ for the positive outcomes and $-[\phi(\mathbf{x}_j\mathbf{b})/\{1 - \Phi(\mathbf{x}_j\mathbf{b})\}]\mathbf{x}_j$ for the negative outcomes, where ϕ is the normal density.

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

Chester Ittner Bliss (1899–1979) was born in Ohio. He was educated as an entomologist, earning degrees from Ohio State and Columbia, and was employed by the United States Department of Agriculture until 1933. When he lost his job because of the Depression, Bliss then worked with R. A. Fisher in London and at the Institute of Plant Protection in Leningrad before returning to a post at the Connecticut Agricultural Experiment Station in 1938. He was also a lecturer at Yale for 25 years. Among many contributions to biostatistics, his development and application of probit methods to biological problems are outstanding.

References

Aldrich, J. H., and F. D. Nelson. 1984. *Linear Probability, Logit, and Probit Models*. Newbury Park, CA: Sage. https://doi.org/10.4135/9781412984744.

Berkson, J. 1944. Application of the logistic function to bio-assay. *Journal of the American Statistical Association* 39: 357–365. https://doi.org/10.2307/2280041.

Bliss, C. I. 1934a. The method of probits. *Science* 79: 38–39. https://doi.org/10.1126/science.79.2037.38.

———. 1934b. The method of probits—a correction. *Science* 79: 409–410. https://doi.org/10.1126/science.79.2053.409.

Cameron, A. C., and P. K. Trivedi. 2022. *Microeconometrics Using Stata*. 2nd ed. College Station, TX: Stata Press.

Cochran, W. G., and D. J. Finney. 1979. Chester Ittner Bliss 1899–1979. *Biometrics* 35: 715–717.

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

Jones, A. M. 2007. *Applied Econometrics for Health Economists: A Practical Guide*. 2nd ed. Abingdon, UK: Radcliffe.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T.-C. Lee. 1985. *The Theory and Practice of Econometrics*. 2nd ed. New York: Wiley.

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.

Newson, R. B., and M. Falcaro. 2023. Robit regression in Stata. *Stata Journal* 23: 658–682.

Pampel, F. C. 2021. *Logistic Regression: A Primer*. 2nd ed. Thousand Oaks, CA: Sage.

Pedace, R. 2013. *Econometrics for Dummies*. Hoboken, NJ: Wiley.

Pinzon, E. 2016. Effects of nonlinear models with interactions of discrete and continuous variables: Estimating, graphing, and interpreting. *The Stata Blog: Not Elsewhere Classified.* https://blog.stata.com/2016/07/12/effects-for-nonlinear-models-with-interactions-of-discrete-and-continuous-variables-estimating-graphing-and-interpreting/.

Powers, D. A., and Y. Xie. 2008. *Statistical Methods for Categorical Data Analysis.* 2nd ed. Bingley, UK: Emerald.

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] **probit postestimation** — Postestimation tools for probit

[R] **biprobit** — Bivariate probit regression

[R] **brier** — Brier score decomposition

[R] **glm** — Generalized linear models

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

[R] **hetprobit** — Heteroskedastic probit model

[R] **ivprobit** — Probit model with continuous endogenous covariates

[R] **logistic** — Logistic regression, reporting odds ratios

[R] **logit** — Logistic regression, reporting coefficients

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

[R] **npregress kernel** — Nonparametric kernel regression

[R] **npregress series** — Nonparametric series regression

[R] **roc** — Receiver operating characteristic (ROC) analysis

[R] **scobit** — Skewed logistic regression

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

[CM] **cmmprobit** — Multinomial probit choice model

[ERM] **eprobit** — Extended probit regression

[FMM] **fmm: probit** — Finite mixtures of probit regression models

[LASSO] **Lasso intro** — Introduction to lasso

[ME] **meprobit** — Multilevel mixed-effects probit regression

[MI] **Estimation** — Estimation commands for use with mi estimate

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

[XT] **xtprobit** — Random-effects and population-averaged probit models

[U] 20 Estimation and postestimation commands