Title

stata.com

glogit — Logit and probit regression for grouped data

Syntax

Logistic regression for grouped data

`blogit` *pos_var pop_var* [*indepvars*] [*if*] [*in*] [, *blogit_options*]

Probit regression for grouped data

`bprobit` *pos_var pop_var* [*indepvars*] [*if*] [*in*] [, *bprobit_options*]

Weighted least-squares logistic regression for grouped data

`glogit` *pos_var pop_var* [*indepvars*] [*if*] [*in*] [, *glogit_options*]

Weighted least-squares probit regression for grouped data

`gprobit` *pos_var pop_var* [*indepvars*] [*if*] [*in*] [, *gprobit_options*]

blogit_options	Description
Model	
`noconstant`	suppress constant term
`asis`	retain perfect predictor variables
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
`collinear`	keep collinear variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`or`	report odds ratios
`nocnsreport`	do not display constraints
display_options	control 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 coefficient table; seldom used
`coeflegend`	display legend instead of statistics

bprobit_options	Description
Model	
noconstant	suppress constant term
asis	retain perfect predictor variables
offset(*varname*)	include *varname* in model with coefficient constrained to 1
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
nocnsreport	do not display constraints
display_options	control 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 coefficient table; seldom used
coeflegend	display legend instead of statistics

glogit_options	Description
SE	
vce(*vcetype*)	*vcetype* may be ols, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
or	report odds ratios
display_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

gprobit_options	Description
SE	
vce(*vcetype*)	*vcetype* may be ols, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
display_options	control column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

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

`bootstrap`, `by`, `jackknife`, `rolling`, and `statsby` are allowed; see [U] **11.1.10 Prefix commands**. `fp` is allowed with `blogit` and `bprobit`.

`nocoef` and `coeflegend` do not appear in the dialog box.

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

Menu

blogit

Statistics > Binary outcomes > Grouped data > Logit regression for grouped data

bprobit

Statistics > Binary outcomes > Grouped data > Probit regression for grouped data

glogit

Statistics > Binary outcomes > Grouped data > Weighted least-squares logit regression

gprobit

Statistics > Binary outcomes > Grouped data > Weighted least-squares probit regression

Description

`blogit` and `bprobit` produce maximum-likelihood logit and probit estimates on grouped ("blocked") data; `glogit` and `gprobit` produce weighted least-squares estimates. In the syntax diagrams above, *pos_var* and *pop_var* refer to variables containing the total number of positive responses and the total population.

See [R] **logistic** for a list of related estimation commands.

Options for blogit and bprobit

Model

`noconstant`; see [R] **estimation options**.

`asis` forces retention of perfect predictor variables and their associated perfectly predicted observations and may produce instabilities in maximization; see [R] **probit**.

`offset(`*varname*`)`, `constraints(`*constraints*`)`, `collinear`; 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`), 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**.

or (blogit only) reports the estimated coefficients transformed to odds ratios, that is, e^b rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or may be specified at estimation or when replaying previously estimated results.

nocnsreport; see [R] **estimation options**.

display_options: 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 blogit and bprobit but are not shown in the dialog box:

nocoef specifies that the coefficient table not be displayed. This option is sometimes used by program writers but is useless interactively.

coeflegend; see [R] **estimation options**.

Options for glogit and gprobit

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (ols) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [R] ***vce_option***.

vce(ols), the default, uses the standard variance estimator for ordinary least-squares regression.

Reporting

level(#); see [R] **estimation options**.

or (glogit only) reports the estimated coefficients transformed to odds ratios, that is, e^b rather than b. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or may be specified at estimation or when replaying previously estimated results.

display_options: noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **estimation options**.

The following option is available with glogit and gprobit but is not shown in the dialog box:

coeflegend; see [R] **estimation options**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Maximum likelihood estimates
Weighted least-squares estimates

Maximum likelihood estimates

`blogit` produces the same results as `logit` and `logistic`, and `bprobit` produces the same results as `probit`, but the "blocked" commands accept data in a slightly different "shape". Consider the following two datasets:

```
. use http://www.stata-press.com/data/r13/xmpl1
. list, sepby(agecat)
```

	agecat	exposed	died	pop
1.	0	0	0	115
2.	0	0	1	5
3.	0	1	0	98
4.	0	1	1	8
5.	1	0	0	69
6.	1	0	1	16
7.	1	1	0	76
8.	1	1	1	22

```
. use http://www.stata-press.com/data/r13/xmpl2
. list
```

	agecat	exposed	deaths	pop
1.	0	0	5	120
2.	0	1	8	106
3.	1	0	16	85
4.	1	1	22	98

These two datasets contain the same information; observations 1 and 2 of `xmpl1` correspond to observation 1 of `xmpl2`, observations 3 and 4 of `xmpl1` correspond to observation 2 of `xmpl2`, and so on.

The first observation of `xmpl1` says that for `agecat==0` and `exposed==0`, 115 subjects did not die (`died==0`). The second observation says that for the same `agecat` and `exposed` groups, five subjects did die (`died==1`). In `xmpl2`, the first observation says that there were five deaths of a population of 120 in `agecat==0` and `exposed==0`. These are two different ways of saying the same thing. Both datasets are transcriptions from the following table, reprinted in Rothman, Greenland, and Lash (2008, 260), for age-specific deaths from all causes for tolbutamide and placebo treatment groups (University Group Diabetes Program 1970):

	Age through 54		Age 55 and above	
	Tolbutamide	Placebo	Tolbutamide	Placebo
Dead	8	5	22	16
Surviving	98	115	76	79

The data in `xmpl1` are said to be "fully relational", which is computer jargon meaning that each observation corresponds to one cell of the table. Stata typically prefers data in this format. The second form of storing these data in `xmpl2` is said to be "folded", which is computer jargon for something less than fully relational.

`blogit` and `bprobit` deal with "folded" data and produce the same results that `logit` and `probit` would have if the data had been stored in the "fully relational" representation.

Example 1

For the tolbutamide data, the fully relational representation is preferred. We could then use `logistic`, `logit`, and any of the epidemiological table commands; see [R] **logistic**, [R] **logit**, and [ST] **epitab**. Nevertheless, there are occasions when the folded representation seems more natural. With `blogit` and `bprobit`, we avoid the tedium of having to unfold the data:

```
. use http://www.stata-press.com/data/r13/xmpl2
. blogit deaths pop agecat exposed, or
Logistic regression for grouped data              Number of obs   =        409
                                                  LR chi2(2)      =      22.47
                                                  Prob > chi2     =     0.0000
Log likelihood =  -142.6212                       Pseudo R2       =     0.0730

------------------------------------------------------------------------------
    _outcome | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      agecat |   4.216299   1.431519     4.24   0.000     2.167361    8.202223
     exposed |   1.404674   .4374454     1.09   0.275     .7629451    2.586175
       _cons |   .0513818   .0170762    -8.93   0.000     .0267868    .0985593
------------------------------------------------------------------------------
```

If we had not specified the `or` option, results would have been presented as coefficients instead of as odds ratios. The estimated odds ratio of death for tolbutamide exposure is 1.40, although the 95% confidence interval includes 1. (By comparison, these data, in fully relational form and analyzed using the `cs` command [see [ST] **epitab**], produce a Mantel–Haenszel weighted odds ratio of 1.40 with a 95% confidence interval of 0.76 to 2.59.)

We can see the underlying coefficients by replaying the estimation results and not specifying the `or` option:

```
. blogit
Logistic regression for grouped data              Number of obs   =        409
                                                  LR chi2(2)      =      22.47
                                                  Prob > chi2     =     0.0000
Log likelihood =  -142.6212                       Pseudo R2       =     0.0730

------------------------------------------------------------------------------
    _outcome |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      agecat |   1.438958   .3395203     4.24   0.000     .7735101    2.104405
     exposed |   .3398053   .3114213     1.09   0.275    -.2705692    .9501798
       _cons |  -2.968471     .33234    -8.93   0.000    -3.619846   -2.317097
------------------------------------------------------------------------------
```

▷ Example 2

bprobit works like blogit, substituting the probit for the logit-likelihood function.

```
. bprobit deaths pop agecat exposed
Probit regression for grouped data                Number of obs   =        409
                                                  LR chi2(2)      =      22.58
                                                  Prob > chi2     =     0.0000
Log likelihood = -142.56478                       Pseudo R2       =     0.0734

------------------------------------------------------------------------------
    _outcome |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      agecat |   .7542049   .1709692     4.41   0.000     .4191114    1.089298
     exposed |   .1906236   .1666059     1.14   0.253    -.1359179    .5171651
       _cons |  -1.673973   .1619594   -10.34   0.000    -1.991408   -1.356539
------------------------------------------------------------------------------
```

◁

Weighted least-squares estimates

▷ Example 3

We have state data for the United States on the number of marriages (marriage), the total population aged 18 years or more (pop18p), and the median age (medage). The dataset excludes Nevada, so it has 49 observations. We now wish to estimate a logit equation for the marriage rate. We will include age squared by specifying the term c.medage#c.medage:

```
. use http://www.stata-press.com/data/r13/census7
(1980 Census data by state)
. glogit marriage pop18p medage c.medage#c.medage
Weighted LS logistic regression for grouped data

      Source |       SS       df       MS              Number of obs =      49
-------------+------------------------------           F(  2,    46) =   12.89
       Model |  .71598314      2  .35799157             Prob > F      =  0.0000
    Residual |  1.27772858    46  .027776708            R-squared     =  0.3591
-------------+------------------------------           Adj R-squared =  0.3313
       Total |  1.99371172    48  .041535661            Root MSE      =  .16666

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      medage |  -.6459349   .2828381    -2.28   0.027    -1.215258   -.0766114
             |
   c.medage# |
   c.medage |   .0095414   .0046608     2.05   0.046     .0001598    .0189231
             |
       _cons |   6.503833   4.288977     1.52   0.136    -2.129431    15.1371
------------------------------------------------------------------------------
```

◁

▷ Example 4

We could just as easily have fit a grouped-probit model by typing `gprobit` rather than `glogit`:

```
. gprobit marriage pop18p medage c.medage#c.medage
Weighted LS probit regression for grouped data
```

Source	SS	df	MS		
Model	.108222962	2	.054111481	Number of obs	= 49
Residual	.192322476	46	.004180923	F(2, 46)	= 12.94
				Prob > F	= 0.0000
				R-squared	= 0.3601
				Adj R-squared	= 0.3323
Total	.300545438	48	.006261363	Root MSE	= .06466

	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
medage	-.2755007	.1121042	-2.46	0.018	-.5011548	-.0498466
c.medage# c.medage	.0041082	.0018422	2.23	0.031	.0004001	.0078163
_cons	2.357708	1.704446	1.38	0.173	-1.073164	5.788579

◁

Stored results

`blogit` and `bprobit` store 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-squared
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(p)`	significance of 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)	blogit or bprobit
e(cmdline)	command as typed
e(depvar)	variable containing number of positive responses and variable containing population size
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(predict)	program used to implement predict
e(marginsok)	predictions allowed 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

glogit and gprobit store the following in e():

Scalars

e(N)	number of observations
e(mss)	model sum of squares
e(df_m)	model degrees of freedom
e(rss)	residual sum of squares
e(df_r)	residual degrees of freedom
e(r2)	R-squared
e(r2_a)	adjusted R-squared
e(F)	F statistic
e(rmse)	root mean squared error
e(rank)	rank of e(V)

Macros

e(cmd)	glogit or gprobit
e(cmdline)	command as typed
e(depvar)	variable containing number of positive responses and variable containing population size
e(model)	ols
e(title)	title in estimation output
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(properties)	b V
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators

Functions

e(sample)	marks estimation sample

Methods and formulas

Methods and formulas are presented under the following headings:

Maximum likelihood estimates

The results reported by `blogit` and `bprobit` are obtained by maximizing a weighted logit- or probit-likelihood function. Let $F(\,)$ denote the normal- or logistic-likelihood function. The likelihood of observing each observation in the data is then

$$F(\beta x)^s\{1-F(\beta x)\}^{t-s}$$

where s is the number of successes and t is the population. The term above is counted as contributing $s+(t-s)=t$ degrees of freedom. All of this follows directly from the definitions of logit and probit.

`blogit` and `bprobit` support 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*.

Weighted least-squares estimates

The logit function is defined as the log of the odds ratio. If there is one explanatory variable, the model can be written as

$$\log\left(\frac{p_j}{1-p_j}\right)=\beta_0+\beta_1 x_j+\epsilon_j \tag{1}$$

where p_j represents successes divided by population for the jth observation. (If there is more than one explanatory variable, we simply interpret β_1 as a row vector and x_j as a column vector.) The large-sample expectation of ϵ_j is zero, and its variance is

$$\sigma_j^2=\frac{1}{n_j p_j(1-p_j)}$$

where n_j represents the population for observation j. We can thus apply weighted least squares to the observations, with weights proportional to $n_j p_j(1-p_j)$.

As in any feasible generalized least-squares problem, estimation proceeds in two steps. First, we fit (1) by OLS and compute the predicted probabilities as

$$\widehat{p_j}=\frac{\exp(\widehat{\beta_0}+\widehat{\beta_1}x_j)}{1+\exp(\widehat{\beta_0}+\widehat{\beta_1}x_j)}$$

In the second step, we fit (1) by using analytic weights equal to $n_j\widehat{p_j}(1-\widehat{p_j})$.

For gprobit, write $\Phi(\cdot)$ for the cumulative normal distribution, and define z_j implicitly by $\Phi(z_j) = p_j$, where p_j is the fraction of successes for observation j. The probit model for one explanatory variable can be written as

$$\Phi^{-1}(p_j) = \beta_0 + \beta_1 x_j + \epsilon_j$$

(If there is more than one explanatory variable, we simply interpret β_1 as a row vector and x_j as a column vector.)

The expectation of ϵ_j is zero, and its variance is given by

$$\sigma_j^2 = \frac{p_j(1-p_j)}{n_j \phi^2\{\Phi^{-1}(p_j)\}}$$

where $\phi(\cdot)$ represents the normal density (Amemiya 1981, 1498). We can thus apply weighted least squares to the observations with weights proportional to $1/\sigma_j^2$. As for grouped logit, we use a two-step estimator to obtain the weighted least-squares estimates.

References

Amemiya, T. 1981. Qualitative response models: A survey. *Journal of Economic Literature* 19: 1483–1536.

Hosmer, D. W., Jr., S. A. Lemeshow, and R. X. Sturdivant. 2013. *Applied Logistic Regression*. 3rd ed. Hoboken, NJ: Wiley.

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.

Rothman, K. J., S. Greenland, and T. L. Lash. 2008. *Modern Epidemiology*. 3rd ed. Philadelphia: Lippincott Williams & Wilkins.

University Group Diabetes Program. 1970. A study of the effects of hypoglycemic agents on vascular complications in patients with adult-onset diabetes, II: Mortality results. *Diabetes* 19, supplement 2: 789–830.

Also see

[R] **glogit postestimation** — Postestimation tools for glogit, gprobit, blogit, and bprobit

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

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

[R] **probit** — Probit regression

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

[U] **20 Estimation and postestimation commands**