cloglog — Complementary log-log regression

Syntax

`cloglog` *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
`collinear`	keep collinear variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `opg`, `bootstrap`, or `jackknife`
Reporting	
`level(`*#*`)`	set confidence level; default is `level(95)`
`eform`	report exponentiated coefficients
`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
`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**.

`bootstrap`, `by`, `fp`, `jackknife`, `mi estimate`, `nestreg`, `rolling`, `statsby`, `stepwise`, and `svy` are allowed; see **[U] 11.1.10 Prefix commands**.

`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**.

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

`coeflegend` does not appear in the dialog box.

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

Menu

Statistics > Binary outcomes > Complementary log-log regression

Description

cloglog fits maximum-likelihood complementary log-log models.

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

Options

Model

noconstant, offset(*varname*); 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**.

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, 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**.

eform displays the exponentiated coefficients and corresponding standard errors and confidence intervals.

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.

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

The following option is available with cloglog 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:

Introduction to complementary log-log regression

`cloglog` fits maximum likelihood models with dichotomous dependent variables coded as 0/1 (or, more precisely, coded as 0 and not 0).

▷ Example 1

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

```
. use http://www.stata-press.com/data/r13/auto
(1978 Automobile Data)

. keep make mpg weight foreign

. describe

Contains data from http://www.stata-press.com/data/r13/auto.dta
  obs:            74                          1978 Automobile Data
 vars:             4                          13 Apr 2013 17:45
 size:         1,702                          (_dta has notes)
───────────────────────────────────────────────────────────────────────────────
              storage   display    value
variable 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 type
───────────────────────────────────────────────────────────────────────────────
Sorted by:  foreign
     Note:  dataset has changed since last saved

. inspect foreign

foreign:  Car type                          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 variable `foreign` 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

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

where $F(z) = 1 - \exp\big\{-\exp(z)\big\}$.

To fit this model, we type

```
. cloglog foreign weight mpg
Iteration 0:   log likelihood = -34.054593
Iteration 1:   log likelihood = -27.869915
Iteration 2:   log likelihood = -27.742997
Iteration 3:   log likelihood = -27.742769
Iteration 4:   log likelihood = -27.742769
Complementary log-log regression                Number of obs     =         74
                                                Zero outcomes     =         52
                                                Nonzero outcomes  =         22
                                                LR chi2(2)        =      34.58
Log likelihood = -27.742769                     Prob > chi2       =     0.0000

------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0029153   .0006974    -4.18   0.000    -.0042823   -.0015483
         mpg |  -.1422911    .076387    -1.86   0.062    -.2920069    .0074247
       _cons |   10.09694   3.351841     3.01   0.003     3.527448    16.66642
------------------------------------------------------------------------------
```

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 when 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, 0 is interpreted as failure and 1 as success. If your dependent variable takes on the values 0, 1, and 2, 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 `cloglog` *y x*, Stata fits the model

$$\Pr(y_j \neq 0 \mid \mathbf{x}_j) = 1 - \exp\Big\{-\exp(\mathbf{x}_j\boldsymbol{\beta})\Big\}$$

❑

Robust standard errors

If you specify the `vce(robust)` option, `cloglog` reports robust standard errors, as described in [U] **20.21 Obtaining robust variance estimates**. For the model of `foreign` on `weight` and `mpg`, the robust calculation increases the standard error of the coefficient on `mpg` by 44%:

```
. cloglog foreign weight mpg, vce(robust)
Iteration 0:   log pseudolikelihood = -34.054593
Iteration 1:   log pseudolikelihood = -27.869915
Iteration 2:   log pseudolikelihood = -27.742997
Iteration 3:   log pseudolikelihood = -27.742769
Iteration 4:   log pseudolikelihood = -27.742769
Complementary log-log regression                Number of obs     =         74
                                                Zero outcomes     =         52
                                                Nonzero outcomes  =         22
                                                Wald chi2(2)      =      29.74
Log pseudolikelihood = -27.742769               Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |               Robust
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0029153   .0007484    -3.90   0.000    -.0043822   -.0014484
         mpg |  -.1422911   .1102466    -1.29   0.197    -.3583704    .0737882
       _cons |   10.09694   4.317305     2.34   0.019     1.635174    18.5587
------------------------------------------------------------------------------
```

Without vce(robust), the standard error for the coefficient on mpg was reported to be 0.076, with a resulting confidence interval of $[-0.29, 0.01]$.

The vce(cluster *clustvar*) option can relax the independence assumption required by the complementary log-log estimator to being just independence between clusters. To demonstrate this ability, we will switch to a different dataset.

We are studying unionization of women in the United States by using the union dataset; see [XT] **xt**. We fit the following model, ignoring that women are observed an average of 5.9 times each in this dataset:

```
. use http://www.stata-press.com/data/r13/union, clear
(NLS Women 14-24 in 1968)
. cloglog union age grade not_smsa south##c.year
Iteration 0:   log likelihood = -13606.373
Iteration 1:   log likelihood = -13540.726
Iteration 2:   log likelihood = -13540.607
Iteration 3:   log likelihood = -13540.607
Complementary log-log regression                Number of obs     =      26200
                                                Zero outcomes     =      20389
                                                Nonzero outcomes  =       5811
                                                LR chi2(6)        =     647.24
Log likelihood = -13540.607                     Prob > chi2       =     0.0000

------------------------------------------------------------------------------
       union |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0185346   .0043616     4.25   0.000      .009986    .0270833
       grade |   .0452772   .0057125     7.93   0.000     .0340809    .0564736
    not_smsa |  -.1886592   .0317801    -5.94   0.000    -.2509471   -.1263712
     1.south |  -1.422292   .3949381    -3.60   0.000    -2.196356    -.648227
        year |  -.0133007   .0049576    -2.68   0.007    -.0230174   -.0035839
             |
south#c.year |
           1 |   .0105659   .0049234     2.15   0.032     .0009161    .0202157
             |
       _cons |  -1.219801   .2952374    -4.13   0.000    -1.798455   -.6411462
------------------------------------------------------------------------------
```

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 different time trend for the south. The `vce(cluster` *clustvar*`)` option provides a way to fit this model and obtains correct standard errors:

```
. cloglog union age grade not_smsa south##c.year, vce(cluster id) nolog
Complementary log-log regression                Number of obs     =      26200
                                                Zero outcomes     =      20389
                                                Nonzero outcomes  =       5811

                                                Wald chi2(6)      =     160.76
Log pseudolikelihood = -13540.607               Prob > chi2       =     0.0000

                              (Std. Err. adjusted for 4434 clusters in idcode)
```

union	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	.0185346	.0084873	2.18	0.029	.0018999	.0351694
grade	.0452772	.0125776	3.60	0.000	.0206255	.069929
not_smsa	-.1886592	.0642068	-2.94	0.003	-.3145021	-.0628162
1.south	-1.422292	.506517	-2.81	0.005	-2.415047	-.4295365
year	-.0133007	.0090628	-1.47	0.142	-.0310633	.004462
south#c.year						
1	.0105659	.0063175	1.67	0.094	-.0018162	.022948
_cons	-1.219801	.5175129	-2.36	0.018	-2.234107	-.2054942

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

```
. xtcloglog union age grade not_smsa south##c.year, pa nolog
GEE population-averaged model                   Number of obs      =     26200
Group variable:                     idcode      Number of groups   =      4434
Link:                              cloglog      Obs per group: min =         1
Family:                           binomial                     avg =       5.9
Correlation:                  exchangeable                     max =        12
                                                Wald chi2(6)       =    234.66
Scale parameter:                         1      Prob > chi2        =    0.0000
```

union	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	.0153737	.0081156	1.89	0.058	-.0005326	.03128
grade	.0549518	.0095093	5.78	0.000	.0363139	.0735897
not_smsa	-.1045232	.0431082	-2.42	0.015	-.1890138	-.0200326
1.south	-1.714868	.3384558	-5.07	0.000	-2.378229	-1.051507
year	-.0115881	.0084125	-1.38	0.168	-.0280763	.0049001
south#c.year						
1	.0149796	.0041687	3.59	0.000	.0068091	.0231501
_cons	-1.488278	.4468005	-3.33	0.001	-2.363991	-.6125652

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

In addition to this estimator, we may use the `xtgee` command to fit a panel estimator (with complementary log-log link) and any number of assumptions on the within-idcode correlation.

`cloglog, 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 (as do the `xtgee` population-averaged models).

Stored results

`cloglog` 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_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(N_f)`	number of zero outcomes
`e(N_s)`	number of nonzero outcomes
`e(df_m)`	model degrees of freedom
`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
`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)`	`cloglog`
`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(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

Methods and formulas

Complementary log-log analysis (related to the gompit model, so named because of its relationship to the Gompertz distribution) is an alternative to logit and probit analysis, but it is unlike these other estimators in that the transformation is not symmetric. Typically, this model is used when the positive (or negative) outcome is rare.

The log-likelihood function for complementary log-log is

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

where S is the set of all observations j such that $y_j \neq 0$, $F(z) = 1 - \exp\big\{-\exp(z)\big\}$, and w_j denotes the optional weights. $\ln L$ is maximized as described in [R] **maximize**.

We can fit a gompit model by reversing the success–failure sense of the dependent variable and using `cloglog`.

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 $\mathbf{u}_j = [\exp(\mathbf{x}_j\mathbf{b}) \exp\big\{-\exp(\mathbf{x}_j\mathbf{b})\big\}/F(\mathbf{x}_j\mathbf{b})]\mathbf{x}_j$ for the positive outcomes and $\{-\exp(\mathbf{x}_j\mathbf{b})\}\mathbf{x}_j$ for the negative outcomes.

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

Acknowledgment

We thank Joseph Hilbe of Arizona State University for providing the inspiration for the `cloglog` command (Hilbe 1996, 1998).

References

Clayton, D. G., and M. Hills. 1993. *Statistical Models in Epidemiology*. Oxford: Oxford University Press.

Hilbe, J. M. 1996. sg53: Maximum-likelihood complementary log-log regression. *Stata Technical Bulletin* 32: 19–20. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, pp. 129–131. College Station, TX: Stata Press.

——. 1998. sg53.2: Stata-like commands for complementary log-log regression. *Stata Technical Bulletin* 41: 23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 7, pp. 166–167. College Station, TX: Stata Press.

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.

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] **cloglog postestimation** — Postestimation tools for cloglog

[R] **clogit** — Conditional (fixed-effects) logistic regression

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

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

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

[ME] **mecloglog** — Multilevel mixed-effects complementary log-log regression

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

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

[XT] **xtcloglog** — Random-effects and population-averaged cloglog models

[U] 20 Estimation and postestimation commands