Title

logit — Logistic regression, reporting coefficients

Syntax

`logit` *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*, `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

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`, `mfp`, `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()`, `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` and `coeflegend` do not appear in the dialog box.

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

Menu

Statistics > Binary outcomes > Logistic regression

Description

logit fits a logit model for a binary response by maximum likelihood; it models the probability of a positive outcome given a set of regressors. *depvar* equal to nonzero and nonmissing (typically *depvar* equal to one) indicates a positive outcome, whereas *depvar* equal to zero indicates a negative outcome.

Also see [R] **logistic**; logistic displays estimates as odds ratios. Many users prefer the logistic command to logit. Results are the same regardless of which you use—both are the maximum-likelihood estimator. Several auxiliary commands that can be run after logit, probit, or logistic estimation are described in [R] **logistic postestimation**. A list of related estimation commands is given in [R] **logistic**.

If estimating on grouped data, see [R] **glogit**.

Options

Model

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

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 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 `logit` 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 of no use interactively.

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

Remarks and examples

Remarks are presented under the following headings:

Basic usage

`logit` fits maximum likelihood models with dichotomous dependent (left-hand-side) 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 logit model explaining whether a car is foreign on the basis of 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 $F(z) = e^z/(1 + e^z)$ is the cumulative logistic distribution.

To fit this model, we type

```
. logit foreign weight mpg
Iteration 0:   log likelihood =  -45.03321
Iteration 1:   log likelihood = -29.238536
Iteration 2:   log likelihood = -27.244139
Iteration 3:   log likelihood = -27.175277
Iteration 4:   log likelihood = -27.175156
Iteration 5:   log likelihood = -27.175156
Logistic regression                               Number of obs   =         74
                                                  LR chi2(2)      =      35.72
                                                  Prob > chi2     =     0.0000
Log likelihood = -27.175156                       Pseudo R2       =     0.3966
```

foreign	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
weight	-.0039067	.0010116	-3.86	0.000	-.0058894	-.001924
mpg	-.1685869	.0919175	-1.83	0.067	-.3487418	.011568
_cons	13.70837	4.518709	3.03	0.002	4.851859	22.56487

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.

◁

❑ 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 `logit` y x, Stata fits the model

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

❑

Model identification

The `logit` command has one more feature, and it is probably the most useful. `logit` automatically checks the model for identification and, if it is underidentified, drops whatever variables and observations are necessary for estimation to proceed. (`logistic`, `probit`, and `ivprobit` do this as well.)

▷ Example 2

Have you ever fit a logit 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 whenever the independent variable is one. In our data, $\Pr(y = 0 \mid x = 1) = 1$, which means that the logit 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 a lot of independent variables in your model. If you have ever had such difficulties, 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. In the meantime, you have been waiting. The estimates that you finally receive, if you receive any at all, may be nothing more than numerical roundoff.

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

Let's return to our automobile data. Among the variables we have in the data is one called `repair`, which 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 http://www.stata-press.com/data/r13/repair, clear
(1978 Automobile Data)

. tabulate foreign repair

           |              repair
  Car type |         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 logit coefficient must be minus infinity, and that would set most computer programs buzzing.

Let's try Stata on this problem.

```
. logit foreign b3.repair
note: 1.repair != 0 predicts failure perfectly
      1.repair dropped and 10 obs not used
Iteration 0:   log likelihood = -26.992087
Iteration 1:   log likelihood = -22.483187
Iteration 2:   log likelihood = -22.230498
Iteration 3:   log likelihood = -22.229139
Iteration 4:   log likelihood = -22.229138
Logistic regression                               Number of obs   =         48
                                                  LR chi2(1)      =       9.53
                                                  Prob > chi2     =     0.0020
Log likelihood = -22.229138                       Pseudo R2       =     0.1765

------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      repair |
          1  |          0  (empty)
          2  |  -2.197225   .7698003    -2.85   0.004    -3.706005   -.6884436
             |
       _cons |  -1.98e-16   .4714045    -0.00   1.000    -.9239359    .9239359
------------------------------------------------------------------------------
```

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 dropped 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 problems like this. 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, "so-and-so predicts outcome perfectly" and stops. Statistics dictates 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 in this case, 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 would say, "age $>$ 65 predicts failure perfectly". It would then inform us about the fix-up it takes and fit what can be fit of our model.

logit (and logistic, probit, and ivprobit) will also occasionally display messages such as

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

There are two causes for a message like this. The first—and most unlikely—case occurs when a continuous variable (or a combination of a continuous variable with other continuous or dummy variables) is simply a great predictor of the dependent variable. Consider Stata's auto.dta dataset with 6 observations removed.

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

. drop if foreign==0 & gear_ratio > 3.1
(6 observations deleted)

. logit foreign mpg weight gear_ratio, nolog

Logistic regression                               Number of obs   =         68
                                                  LR chi2(3)      =      72.64
                                                  Prob > chi2     =     0.0000
Log likelihood = -6.4874814                       Pseudo R2       =     0.8484

------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |  -.4944907   .2655508    -1.86   0.063    -1.014961    .0259792
      weight |  -.0060919    .003101    -1.96   0.049    -.0121698    -.000014
  gear_ratio |   15.70509   8.166234     1.92   0.054     -.300436    31.71061
       _cons |  -21.39527   25.41486    -0.84   0.400    -71.20747    28.41694
------------------------------------------------------------------------------
Note: 4 failures and 0 successes completely determined.
```

There are no missing standard errors in the output. If you receive the "completely determined" message and have one or more missing standard errors in your output, see the second case discussed below.

Note gear_ratio's large coefficient. logit thought that the 4 observations with the smallest predicted probabilities were essentially predicted perfectly.

```
. predict p
(option pr assumed; Pr(foreign))

. sort p

. list p in 1/4

     +----------+
     |        p |
     |----------|
  1. | 1.34e-10 |
  2. | 6.26e-09 |
  3. | 7.84e-09 |
  4. | 1.49e-08 |
     +----------+
```

If this happens to you, you do not have to do anything. Computationally, the model is sound. The second case discussed below requires careful examination.

The second case occurs when the independent terms are all dummy variables or continuous ones with repeated values (for example, age). Here one or more of the estimated coefficients will have missing standard errors. For example, consider this dataset consisting of 5 observations.

```
. use http://www.stata-press.com/data/r13/logitxmpl, clear
. list, separator(0)

     +-------------+
     | y   x1   x2 |
     |-------------|
  1. | 0    0    0 |
  2. | 0    0    0 |
  3. | 0    1    0 |
  4. | 1    1    0 |
  5. | 0    0    1 |
  6. | 1    0    1 |
     +-------------+

. logit y x1 x2
Iteration 0:   log likelihood =  -3.819085
Iteration 1:   log likelihood = -2.9527336
Iteration 2:   log likelihood = -2.8110282
Iteration 3:   log likelihood = -2.7811973
Iteration 4:   log likelihood = -2.7746107
Iteration 5:   log likelihood = -2.7730128
  (output omitted)
Iteration 15996: log likelihood = -2.7725887  (not concave)
Iteration 15997: log likelihood = -2.7725887  (not concave)
Iteration 15998: log likelihood = -2.7725887  (not concave)
Iteration 15999: log likelihood = -2.7725887  (not concave)
Iteration 16000: log likelihood = -2.7725887  (not concave)
convergence not achieved
Logistic regression                               Number of obs   =          6
                                                  LR chi2(1)      =       2.09
                                                  Prob > chi2     =     0.1480
Log likelihood = -2.7725887                       Pseudo R2       =     0.2740

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |    18.3704          2     9.19   0.000     14.45047    22.29033
          x2 |    18.3704          .        .       .            .           .
       _cons |   -18.3704   1.414214   -12.99   0.000    -21.14221    -15.5986
------------------------------------------------------------------------------
Note: 2 failures and 0 successes completely determined.
convergence not achieved
r(430);
```

Three things are happening here. First, `logit` iterates almost forever and then declares nonconvergence. Second, `logit` can fit the outcome ($y = 0$) for the covariate pattern `x1` $= 0$ and `x2` $= 0$ (that is, the first two observations) perfectly. This observation is the “2 failures and 0 successes completely determined”. Third, if this observation is dropped, then `x1`, `x2`, and the constant are collinear.

This is the cause of the nonconvergence, the message “completely determined”, and the missing standard errors. It happens when you have a covariate pattern (or patterns) with only one outcome and there is collinearity when the observations corresponding to this covariate pattern are dropped.

If this happens to you, confirm the causes. First, identify the covariate pattern with only one outcome. (For your data, replace `x1` and `x2` with the independent variables of your model.)

```
. egen pattern = group(x1 x2)
. quietly logit y x1 x2, iterate(100)
. predict p
(option pr assumed; Pr(y))
. summarize p
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
           p |          6    .3333333    .2581989   1.05e-08         .5
```

If successes were completely determined, that means that there are predicted probabilities that are almost 1. If failures were completely determined, that means that there are predicted probabilities that are almost 0. The latter is the case here, so we locate the corresponding value of `pattern`:

```
. tabulate pattern if p < 1e-7
   group(x1 |
        x2) |      Freq.     Percent        Cum.
------------+-----------------------------------
          1 |          2      100.00      100.00
------------+-----------------------------------
      Total |          2      100.00
```

Once we omit this covariate pattern from the estimation sample, `logit` can deal with the collinearity:

```
. logit y x1 x2 if pattern != 1, nolog
note: x2 omitted because of collinearity
Logistic regression                               Number of obs   =          4
                                                  LR chi2(1)      =       0.00
                                                  Prob > chi2     =     1.0000
Log likelihood = -2.7725887                       Pseudo R2       =     0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |          0           2     0.00   1.000    -3.919928    3.919928
          x2 |          0  (omitted)
       _cons |          0    1.414214     0.00   1.000    -2.771808    2.771808
------------------------------------------------------------------------------
```

We omit the collinear variable. Then we must decide whether to include or omit the observations with `pattern` = 1. We could include them,

```
. logit y x1, nolog
Logistic regression                               Number of obs   =          6
                                                  LR chi2(1)      =       0.37
                                                  Prob > chi2     =     0.5447
Log likelihood = -3.6356349                       Pseudo R2       =     0.0480

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   1.098612    1.825742     0.60   0.547    -2.479776    4.677001
       _cons |  -1.098612    1.154701    -0.95   0.341    -3.361784    1.164559
------------------------------------------------------------------------------
```

or exclude them,

```
. logit y x1 if pattern != 1, nolog
Logistic regression                               Number of obs   =          4
                                                  LR chi2(1)      =       0.00
                                                  Prob > chi2     =     1.0000
Log likelihood = -2.7725887                       Pseudo R2       =     0.0000
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
x1	0	2	0.00	1.000	-3.919928	3.919928
_cons	0	1.414214	0.00	1.000	-2.771808	2.771808

If the covariate pattern that predicts outcome perfectly is meaningful, you may want to exclude these observations from the model. Here you would report that covariate pattern such and such predicted outcome perfectly and that the best model for the rest of the data is .... But, more likely, the perfect prediction was simply the result of having too many predictors in the model. Then you would omit the extraneous variables from further consideration and report the best model for all the data.

❑

Stored results

`logit` 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-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)`	`logit`
`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(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

Methods and formulas

Cramer (2003, chap. 9) surveys the prehistory and history of the logit model. The word "logit" was coined by Berkson (1944) and is analogous to the word "probit". For an introduction to probit and logit, see, for example, Aldrich and Nelson (1984), Cameron and Trivedi (2010), Greene (2012), Jones (2007), Long (1997), Long and Freese (2014), Pampel (2000), or Powers and Xie (2008).

The likelihood function for logit 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 $y_j \neq 0$, $F(z) = e^z/(1+e^z)$, 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 $\mathbf{u}_j = \{1 - F(\mathbf{x}_j\mathbf{b})\}\mathbf{x}_j$ for the positive outcomes and $-F(\mathbf{x}_j\mathbf{b})\mathbf{x}_j$ for the negative outcomes.

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

Joseph Berkson (1899–1982) was born in New York City and studied at the College of the City of New York, Columbia, and Johns Hopkins, earning both an MD and a doctorate in statistics. He then worked at Johns Hopkins before moving to the Mayo Clinic in 1931 as a biostatistician. Among many other contributions, his most influential one drew upon a long-sustained interest in the logistic function, especially his 1944 paper on bioassay, in which he introduced the term “logit”. Berkson was a frequent participant in controversy—sometimes humorous, sometimes bitter—on subjects such as the evidence for links between smoking and various diseases and the relative merits of probit and logit methods and of different calculation methods.

References

Aldrich, J. H., and F. D. Nelson. 1984. *Linear Probability, Logit, and Probit Models*. Newbury Park, CA: Sage.

Archer, K. J., and S. A. Lemeshow. 2006. Goodness-of-fit test for a logistic regression model fitted using survey sample data. *Stata Journal* 6: 97–105.

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

Buis, M. L. 2010a. Direct and indirect effects in a logit model. *Stata Journal* 10: 11–29.

——. 2010b. Stata tip 87: Interpretation of interactions in nonlinear models. *Stata Journal* 10: 305–308.

Cameron, A. C., and P. K. Trivedi. 2010. *Microeconometrics Using Stata*. Rev. ed. College Station, TX: Stata Press.

Cleves, M. A., and A. Tosetto. 2000. sg139: Logistic regression when binary outcome is measured with uncertainty. *Stata Technical Bulletin* 55: 20–23. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 152–156. College Station, TX: Stata Press.

Cramer, J. S. 2003. *Logit Models from Economics and Other Fields*. Cambridge: Cambridge University Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Hilbe, J. M. 2009. *Logistic Regression Models*. Boca Raton, FL: Chapman & Hill/CRC.

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

Jones, A. 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.

Mitchell, M. N., and X. Chen. 2005. Visualizing main effects and interactions for binary logit models. *Stata Journal* 5: 64–82.

O'Fallon, W. M. 1998. Berkson, Joseph. In Vol. 1 of *Encyclopedia of Biostatistics*, ed. P. Armitage and T. Colton, 290–295. Chichester, UK: Wiley.

Orsini, N., R. Bellocco, and P. C. Sjölander. 2013. Doubly robust estimation in generalized linear models. *Stata Journal* 13: 185–205.

Pampel, F. C. 2000. *Logistic Regression: A Primer*. Thousand Oaks, CA: Sage.

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

Pregibon, D. 1981. Logistic regression diagnostics. *Annals of Statistics* 9: 705–724.

Schonlau, M. 2005. Boosted regression (boosting): An introductory tutorial and a Stata plugin. *Stata Journal* 5: 330–354.

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] **logit postestimation** — Postestimation tools for logit

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

[R] **cloglog** — Complementary log-log regression

[R] **exlogistic** — Exact logistic regression

[R] **glogit** — Logit and probit regression for grouped data

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

[R] **probit** — Probit regression

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

[ME] **melogit** — Multilevel mixed-effects logistic regression

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

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

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

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

Title

logit postestimation — Postestimation tools for logit

Description

The following postestimation commands are of special interest after `logit`:

Command	Description
`estat classification`	report various summary statistics, including the classification table
`estat gof`	Pearson or Hosmer–Lemeshow goodness-of-fit test
`lroc`	compute area under ROC curve and graph the curve
`lsens`	graph sensitivity and specificity versus probability cutoff

These commands are not appropriate after the `svy` prefix.

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estat` (svy)	postestimation statistics for survey data
`estimates`	cataloging estimation results
`forecast`[1]	dynamic forecasts and simulations
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`linktest`	link test for model specification
`lrtest`[2]	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`suest`	seemingly unrelated estimation
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

[1] `forecast` is not appropriate with `mi` or `svy` estimation results.

[2] `lrtest` is not appropriate with `svy` estimation results.

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset` `rules` `asif`]

statistic	Description
Main	
`pr`	probability of a positive outcome; the default
`xb`	linear prediction
`stdp`	standard error of the prediction
*`dbeta`	Pregibon (1981) $\Delta\widehat{\beta}$ influence statistic
*`deviance`	deviance residual
*`dx2`	Hosmer, Lemeshow, and Sturdivant (2013) $\Delta\,\chi^2$ influence statistic
*`ddeviance`	Hosmer, Lemeshow, and Sturdivant (2013) $\Delta\,D$ influence statistic
*`hat`	Pregibon (1981) leverage
*`number`	sequential number of the covariate pattern
*`residuals`	Pearson residuals; adjusted for number sharing covariate pattern
*`rstandard`	standardized Pearson residuals; adjusted for number sharing covariate pattern
`score`	first derivative of the log likelihood with respect to $\mathbf{x}_j\boldsymbol{\beta}$

Unstarred statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample. Starred statistics are calculated only for the estimation sample, even when `if e(sample)` is not specified.

`pr`, `xb`, `stdp`, and `score` are the only options allowed with `svy` estimation results.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`pr`, the default, calculates the probability of a positive outcome.

`xb` calculates the linear prediction.

`stdp` calculates the standard error of the linear prediction.

`dbeta` calculates the Pregibon (1981) $\Delta\widehat{\beta}$ influence statistic, a standardized measure of the difference in the coefficient vector that is due to deletion of the observation along with all others that share the same covariate pattern. In Hosmer, Lemeshow, and Sturdivant (2013, 154–155) jargon, this statistic is M-asymptotic; that is, it is adjusted for the number of observations that share the same covariate pattern.

`deviance` calculates the deviance residual.

`dx2` calculates the Hosmer, Lemeshow, and Sturdivant (2013, 191) $\Delta\chi^2$ influence statistic, reflecting the decrease in the Pearson χ^2 that is due to deletion of the observation and all others that share the same covariate pattern.

`ddeviance` calculates the Hosmer, Lemeshow, and Sturdivant (2013, 191) ΔD influence statistic, which is the change in the deviance residual that is due to deletion of the observation and all others that share the same covariate pattern.

`hat` calculates the Pregibon (1981) leverage or the diagonal elements of the hat matrix adjusted for the number of observations that share the same covariate pattern.

`number` numbers the covariate patterns—observations with the same covariate pattern have the same `number`. Observations not used in estimation have `number` set to missing. The first covariate pattern is numbered 1, the second 2, and so on.

`residuals` calculates the Pearson residual as given by Hosmer, Lemeshow, and Sturdivant (2013, 155) and adjusted for the number of observations that share the same covariate pattern.

`rstandard` calculates the standardized Pearson residual as given by Hosmer, Lemeshow, and Sturdivant (2013, 191) and adjusted for the number of observations that share the same covariate pattern.

`score` calculates the equation-level score, $\partial \ln L/\partial(\mathbf{x}_j\boldsymbol{\beta})$.

Options

`nooffset` is relevant only if you specified `offset(`*varname*`)` for `logit`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_j\mathbf{b}$ rather than as $\mathbf{x}_j\mathbf{b} + \text{offset}_j$.

`rules` requests that Stata use any rules that were used to identify the model when making the prediction. By default, Stata calculates missing for excluded observations.

`asif` requests that Stata ignore the rules and exclusion criteria and calculate predictions for all observations possible by using the estimated parameter from the model.

Remarks and examples

Once you have fit a logit model, you can obtain the predicted probabilities by using the `predict` command for both the estimation sample and other samples; see [U] **20 Estimation and postestimation commands** and [R] **predict**. Here we will make only a few more comments.

`predict` without arguments calculates the predicted probability of a positive outcome, that is, $\Pr(y_j = 1) = F(\mathbf{x}_j\mathbf{b})$. With the `xb` option, `predict` calculates the linear combination $\mathbf{x}_j\mathbf{b}$, where $\mathbf{x}_j$ are the independent variables in the jth observation and $\mathbf{b}$ is the estimated parameter vector. This is sometimes known as the index function because the cumulative distribution function indexed at this value is the probability of a positive outcome.

In both cases, Stata remembers any rules used to identify the model and calculates missing for excluded observations, unless `rules` or `asif` is specified. For information about the other statistics available after `predict`, see [R] **logistic postestimation**.

▷ Example 1: Predicted probabilities

In example 2 of [R] **logit**, we fit the logit model `logit foreign b3.repair`. To obtain predicted probabilities, type

```
. use http://www.stata-press.com/data/r13/repair
(1978 Automobile Data)
. logit foreign b3.repair
note: 1.repair != 0 predicts failure perfectly
      1.repair dropped and 10 obs not used
  (output omitted)
. predict p
(option pr assumed; Pr(foreign))
(10 missing values generated)
```

```
. summarize foreign p

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     foreign |         58    .2068966    .4086186          0          1
           p |         48         .25    .1956984         .1         .5
```

Stata remembers any rules used to identify the model and sets predictions to missing for any excluded observations. `logit` dropped the variable `1.repair` from our model and excluded 10 observations. Thus when we typed `predict p`, those same 10 observations were again excluded, and their predictions were set to missing.

`predict`'s `rules` option uses the rules in the prediction. During estimation, we were told "1.repair != 0 predicts failure perfectly", so the rule is that when `1.repair` is not zero, we should predict 0 probability of success or a positive outcome:

```
. predict p2, rules
(option pr assumed; Pr(foreign))

. summarize foreign p p2

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     foreign |         58    .2068966    .4086186          0          1
           p |         48         .25    .1956984         .1         .5
          p2 |         58    .2068966    .2016268          0         .5
```

`predict`'s `asif` option ignores the rules and exclusion criteria and calculates predictions for all observations possible by using the estimated parameters from the model:

```
. predict p3, asif
(option pr assumed; Pr(foreign))

. summarize foreign p p2 p3

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     foreign |         58    .2068966    .4086186          0          1
           p |         48         .25    .1956984         .1         .5
          p2 |         58    .2068966    .2016268          0         .5
          p3 |         58    .2931035    .2016268         .1         .5
```

Which is right? What `predict` does by default is the most conservative approach. If many observations had been excluded because of a simple rule, we could be reasonably certain that the `rules` prediction is correct. The `asif` prediction is correct only if the exclusion is a fluke, and we would be willing to exclude the variable from the analysis anyway. Then, however, we would refit the model to include the excluded observations.

◁

▷ Example 2: Predictive margins

We can use the command `margins, contrast` after `logit` to make comparisons on the probability scale. Let's fit a model predicting low birthweight from characteristics of the mother:

```
. use http://www.stata-press.com/data/r13/lbw, clear
(Hosmer & Lemeshow data)
```

```
. logit low age i.race i.smoke ptl i.ht i.ui
Iteration 0:   log likelihood =   -117.336
Iteration 1:   log likelihood = -103.81846
Iteration 2:   log likelihood = -103.40486
Iteration 3:   log likelihood = -103.40384
Iteration 4:   log likelihood = -103.40384
Logistic regression                               Number of obs     =        189
                                                  LR chi2(7)        =      27.86
                                                  Prob > chi2       =     0.0002
Log likelihood = -103.40384                       Pseudo R2         =     0.1187

------------------------------------------------------------------------------
         low |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -.0403293   .0357127    -1.13   0.259    -.1103249    .0296663
             |
        race |
      black  |   1.009436   .5025122     2.01   0.045     .0245302    1.994342
      other  |   1.001908   .4248342     2.36   0.018     .1692485    1.834568
             |
       smoke |
     smoker  |   .9631876   .3904357     2.47   0.014     .1979477    1.728427
         ptl |   .6288678   .3399067     1.85   0.064    -.0373371    1.295073
        1.ht |   1.358142   .6289555     2.16   0.031      .125412    2.590872
        1.ui |   .8001832   .4572306     1.75   0.080    -.0959724    1.696339
       _cons |  -1.184127   .9187461    -1.29   0.197    -2.984837    .6165818
------------------------------------------------------------------------------
```

The coefficients are log odds-ratios: conditional on the other predictors, smoking during pregnancy is associated with an increase of 0.96 in the log odds-ratios of low birthweight. The model is linear in the log odds-scale, so the estimate of 0.96 has the same interpretation, whatever the values of the other predictors might be. We could convert 0.96 to an odds ratio by replaying the results with `logit, or`.

But what if we want to talk about the probability of low birthweight, and not the odds? Then we will need the command `margins, contrast`. We will use the `r.` contrast operator to compare each level of `smoke` with a reference level. (`smoke` has only two levels, so there will be only one comparison: a comparison of smokers with nonsmokers.)

```
. margins r.smoke, contrast
Contrasts of predictive margins
Model VCE    : OIM
Expression   : Pr(low), predict()

------------------------------------------------
             |         df        chi2     P>chi2
-------------+----------------------------------
       smoke |          1        6.32     0.0119
------------------------------------------------

--------------------------------------------------------------
                     |            Delta-method
                     |   Contrast   Std. Err.     [95% Conf. Interval]
---------------------+----------------------------------------
               smoke |
(smoker vs nonsmoker)  |   .1832779   .0728814      .0404329    .3261229
--------------------------------------------------------------
```

We see that maternal smoking is associated with an 18.3% increase in the probability of low birthweight. (We received a contrast in the probability scale because predicted probabilities are the default when `margins` is used after `logit`.)

The contrast of 18.3% is a difference of margins that are computed by averaging over the predictions for observations in the estimation sample. If the values of the other predictors were different, the contrast for smoke would be different, too. Let's estimate the contrast for 25-year-old mothers:

```
. margins r.smoke, contrast at(age=25)

Contrasts of predictive margins
Model VCE    : OIM

Expression   : Pr(low), predict()
at           : age             =          25
```

	df	chi2	P>chi2
smoke	1	6.19	0.0129

	Contrast	Delta-method Std. Err.	[95% Conf.	Interval]
smoke (smoker vs nonsmoker)	.1808089	.0726777	.0383632	.3232547

Specifying a maternal age of 25 changed the contrast to 18.1%. Our contrast of probabilities changed because the logit model is nonlinear in the probability scale. A contrast of log odds-ratios would not have changed.

◁

Methods and formulas

See *Methods and formulas* of the individual postestimation commands for details.

References

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

Newson, R. B. 2013. Attributable and unattributable risks and fractions and other scenario comparisons. *Stata Journal* 13: 672–698.

Powers, D. A., H. Yoshioka, and M.-S. Yun. 2011. mvdcmp: Multivariate decomposition for nonlinear response models. *Stata Journal* 11: 556–576.

Pregibon, D. 1981. Logistic regression diagnostics. *Annals of Statistics* 9: 705–724.

Also see

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

[R] **estat classification** — Classification statistics and table

[R] **estat gof** — Pearson or Hosmer–Lemeshow goodness-of-fit test

[R] **lroc** — Compute area under ROC curve and graph the curve

[R] **lsens** — Graph sensitivity and specificity versus probability cutoff

[U] 20 Estimation and postestimation commands