Title

stata.com

exlogistic — Exact logistic regression

Description Quick start Menu Syntax
Options Remarks and examples Stored results Methods and formulas
References Also see

Description

`exlogistic` fits an exact logistic regression model, which produces more accurate inference in small samples than the standard maximum-likelihood–based logistic regression estimator. It can also better deal with completely determined outcomes. `exlogistic` with the `group(`*varname*`)` option conditions on the number of positive outcomes within stratum and is an alternative to the conditional (fixed-effects) logistic regression estimator.

Unlike Stata's other estimation commands, `exlogistic` must perform hypothesis tests during estimation rather than during postestimation with standard postestimation commands.

Quick start

Exact logistic regression of `y` on `x1`, `x2`, and `x3`

```
exlogistic y x1 x2 x3
```

As above, but condition on values of `x3` to save time and memory

```
exlogistic y x1 x2, condvars(x3)
```

As above, and allow more memory for computing the conditional distribution of sufficient statistics

```
exlogistic y x1 x2, condvars(x3) memory(100m)
```

Using data stored in binomial form with `ys` successes out of `n` trials

```
exlogistic ys x1 x2 x3, binomial(n)
```

Report coefficients rather than odds ratios

```
exlogistic y x1 x2 x3, coef
```

Report conditional scores tests

```
exlogistic y x1 x2 x3, test(score)
```

As above, and report joint test for `x1` and `x2`

```
exlogistic y x1 x2 x3, test(score) terms(t1=x1 x2)
```

Include strata-specific constant terms for each level of `svar` for an exact version of conditional logistic regression

```
exlogistic y x1 x2 x3, group(svar)
```

Menu

Statistics > Exact statistics > Exact logistic regression

Syntax

```
exlogistic depvar indepvars [if] [in] [weight] [, options]
```

depvar can be specified as a zero or nonzero variable or the number of positive outcomes within each trial. For a zero or nonzero variable, zero indicates failure and nonzero indicates success. To specify *depvar* as the number of positive outcomes, you must also specify `binomial(`*varname*`|`*#*`)`.

options	Description
Model	
`condvars(`*varlist*`)`	condition on variables in *varlist*
`group(`*varname*`)`	groups/strata are stratified by unique values of *varname*
`binomial(`*varname*`|`*#*`)`	data are in binomial form and the number of trials is contained in *varname* or in *#*
`estconstant`	estimate constant term; do not condition on the number of successes
`noconstant`	suppress constant term
Terms	
`terms(`*termsdef*`)`	terms definition
Options	
`memory(#[b|k|m|g])`	set limit on memory usage; default is `memory(10m)`
`saving(`*filename*`)`	save the joint conditional distribution to *filename*
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`coef`	report estimated coefficients
`test(`*testopt*`)`	report *p*-value for observed sufficient statistic, conditional scores test, or conditional probabilities test
`mue(`*varlist*`)`	compute the median unbiased estimates for *varlist*
`midp`	use the mid-*p*-value rule
`nolog`	do not display the enumeration log

`by`, `statsby`, and `xi` are allowed; see [U] **11.1.10 Prefix commands**.

`fweights` are allowed; see [U] **11.1.6 weight**.

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

Options

Model

`condvars(`*varlist*`)` specifies variables whose parameter estimates are not of interest to you. You can save substantial computer time and memory moving such variables from *indepvars* to `condvars()`. Understand that you will get the same results for `x1` and `x3` whether you type

```
. exlogistic y x1 x2 x3 x4
```

or

```
. exlogistic y x1 x3, condvars(x2 x4)
```

`group(`*varname*`)` specifies the variable defining the strata, if any. A constant term is assumed for each stratum identified in *varname*, and the sufficient statistics for *indepvars* are conditioned on the observed number of successes within each group. This makes the model estimated equivalent to that estimated by `clogit`, Stata's conditional logistic regression command (see [R] **clogit**). `group()` may not be specified with `noconstant` or `estconstant`.

`binomial(`*varname* | *#*`)` indicates that the data are in binomial form and *depvar* contains the number of successes. *varname* contains the number of trials for each observation. If all observations have the same number of trials, you can instead specify the number as an integer. The number of trials must be a positive integer at least as great as the number of successes. If `binomial()` is not specified, the data are assumed to be Bernoulli, meaning that *depvar* equaling zero or nonzero records one failure or success.

`estconstant` estimates the constant term. By default, the models are assumed to have an intercept (constant), but the value of the intercept is not calculated. That is, the conditional distribution of the sufficient statistics for the *indepvars* is computed given the number of successes in *depvar*, thus conditioning out the constant term of the model. Use `estconstant` if you want the estimate of the intercept reported. `estconstant` may not be specified with `group()`.

`noconstant`; see [R] **estimation options**. `noconstant` may not be specified with `group()`.

Terms

`terms(`*termname* = *variable* ... *variable* [`,` *termname* = *variable* ... *variable* ...]`)` defines additional terms of the model on which you want `exlogistic` to perform joint-significance hypothesis tests. By default, `exlogistic` reports tests individually on each variable in *indepvars*. For instance, if variables `x1` and `x3` are in *indepvars*, and you want to jointly test their significance, specify `terms(t1=x1 x3)`. To also test the joint significance of `x2` and `x4`, specify `terms(t1=x1 x3, t2=x2 x4)`. Each variable can be assigned to only one term.

Joint tests are computed only for the conditional scores tests and the conditional probabilities tests. See the `test()` option below.

Options

`memory(#[ b | k | m | g ])` sets a limit on the amount of memory `exlogistic` can use when computing the conditional distribution of the parameter sufficient statistics. The default is `memory(10m)`, where `m` stands for megabyte, or 1,048,576 bytes. The following are also available: `b` stands for byte; `k` stands for kilobyte, which is equal to 1,024 bytes; and `g` stands for gigabyte, which is equal to 1,024 megabytes. The minimum setting allowed is `1m` and the maximum is `2048m` or `2g`, but do not attempt to use more memory than is available on your computer. Also see the first technical note under example 4 on counting the conditional distribution.

`saving(`*filename* [`, replace`]`)` saves the joint conditional distribution to *filename*. This distribution is conditioned on those variables specified in `condvars()`. Use `replace` to replace an existing file with *filename*. A Stata data file is created containing all the feasible values of the parameter sufficient statistics. The variable names are the same as those in *indepvars*, in addition to a variable named `_f_` containing the feasible value frequencies (sometimes referred to as the condition numbers).

Reporting

`level(#)`; see [R] **estimation options**. The `level(#)` option will not work on replay because confidence intervals are based on estimator-specific enumerations. To change the confidence level, you must refit the model.

`coef` reports the estimated coefficients rather than odds ratios (exponentiated coefficients). `coef` may be specified when the model is fit or upon replay. `coef` affects only how results are displayed and not how they are estimated.

`test(sufficient | score | probability)` reports the p-value associated with the observed sufficient statistics, the conditional scores tests, or the conditional probabilities tests, respectively. The default is `test(sufficient)`. If `terms()` is included in the specification, the conditional scores test and the conditional probabilities test are applied to each term providing conditional inference for several parameters simultaneously. All the statistics are computed at estimation time regardless of which is specified. Each statistic may thus also be displayed postestimation without having to refit the model; see [R] **exlogistic postestimation**.

`mue(`*varlist*`)` specifies that median unbiased estimates (MUEs) be reported for the variables in *varlist*. By default, the conditional maximum likelihood estimates (CMLEs) are reported, except for those parameters for which the CMLEs are infinite. Specify `mue(_all)` if you want MUEs for all the *indepvars*.

`midp` instructs `exlogistic` to use the mid-p-value rule when computing the MUEs, p-values, and confidence intervals. This adjustment is for the discreteness of the distribution and halves the value of the discrete probability of the observed statistic before adding it to the p-value. The mid-p-value rule cannot be applied to MUEs whose corresponding parameter CMLE is infinite.

`nolog` prevents the display of the enumeration log. By default, the enumeration log is displayed, showing the progress of computing the conditional distribution of the sufficient statistics.

Remarks and examples

stata.com

Exact logistic regression is the estimation of the logistic model parameters by using the conditional distribution of the parameter sufficient statistics. The estimates are referred to as the conditional maximum likelihood estimates (CMLEs). This technique was first introduced by Cox and Snell (1989) as an alternative to using maximum likelihood estimation, which can perform poorly for small sample sizes. For stratified data, exact logistic regression is a small-sample alternative to conditional logistic regression. See [R] **logit**, [R] **logistic**, and [R] **clogit** to obtain maximum likelihood estimates (MLEs) for the logistic model and the conditional logistic model. For a comprehensive overview of exact logistic regression, see Mehta and Patel (1995).

Let Y_i denote a Bernoulli random variable where we observe the outcome $Y_i = y_i$, $i = 1, \ldots, n$. Associated with each independent observation is a $1 \times p$ vector of covariates, $\mathbf{x}_i$. We will denote $\pi_i = \Pr(Y_i \mid \mathbf{x}_i)$ and let the logit function model the relationship between Y_i and $\mathbf{x}_i$,

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \theta + \mathbf{x}_i\boldsymbol{\beta}$$

where the constant term θ and the $1 \times p$ vector of regression parameters $\boldsymbol{\beta}$ are unknown. The probability of observing $Y_i = y_i$, $i = 1, \ldots, n$, is

$$\Pr(\mathbf{Y} = \mathbf{y}) = \prod_{i=1}^{n} \pi_i^{y_i} (1-\pi_i)^{1-y_i}$$

where $\mathbf{Y} = (Y_1, \ldots, Y_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$. The MLEs for θ and $\boldsymbol{\beta}$ maximize the log of this function.

The sufficient statistics for θ and β_j, $j = 1, \ldots, p$, are $M = \sum_{i=1}^{n} Y_i$ and $T_j = \sum_{i=1}^{n} Y_i x_{ij}$, respectively, and we observe $M = m$ and $T_j = t_j$. By default, `exlogistic` tallies the conditional distribution of $\mathbf{T} = (T_1, \ldots, T_p)$ given $M = m$. This distribution will have a size of $\binom{n}{m}$. (It would have a size of 2^n without conditioning on $M = m$.) Denote one of these vectors $\mathbf{T}^{(k)} = (t_1^{(k)}, \ldots, t_p^{(k)})$, $k = 1, \ldots, N$, with combinatorial coefficient (frequency) c_k, $\sum_{k=1}^{N} c_k = \binom{n}{m}$. For each independent variable x_j, $j = 1, \ldots, p$, we reduce the conditional distribution further by conditioning on all other observed sufficient statistics $T_l = t_l$, $l \neq j$. The conditional probability of observing $T_j = t_j$ has the form

$$\Pr(T_j = t_j \mid T_l = t_l, l \neq j, M = m) = \frac{c\, e^{t_j \beta_j}}{\sum_k c_k e^{t_j^{(k)} \beta_j}}$$

where the sum is over the subset of $\mathbf{T}$ vectors such that $(T_1^{(k)} = t_1, \ldots, T_j^{(k)} = t_j^{(k)}, \ldots, T_p^{(k)} = t_p)$ and c is the combinatorial coefficient associated with the observed $\mathbf{t}$. The CMLE for β_j maximizes the log of this function.

Specifying nuisance variables in `condvars()` will reduce the size of the conditional distribution by conditioning on their observed sufficient statistics as well as conditioning on $M = m$. This reduces the amount of memory consumed at the cost of not obtaining regression estimates for those variables specified in `condvars()`.

Inferences from MLEs rely on asymptotics, and if your sample size is small, these inferences may not be valid. On the other hand, inferences from the CMLEs are exact in the sense that they use the conditional distribution of the sufficient statistics outlined above.

For small datasets, it is common for the dependent variable to be completely determined by the data. Here the MLEs and the CMLEs are unbounded. `exlogistic` will instead compute the MUE, the regression estimate that places the observed sufficient statistic at the median of the conditional distribution.

▷ Example 1

One example presented by Mehta and Patel (1995) is data from a prospective study of perinatal infection and human immunodeficiency virus type 1 (HIV-1). We use a variation of this dataset. There was an investigation Hutto et al. (1991) into whether the blood serum levels of glycoproteins CD4 and CD8 measured in infants at 6 months of age might predict their development of HIV infection. The blood serum levels are coded as ordinal values 0, 1, and 2.

```
. use http://www.stata-press.com/data/r15/hiv1
(prospective study of perinatal infection of HIV-1)
. list in 1/5
```

	hiv	cd4	cd8
1.	1	0	0
2.	0	0	0
3.	1	0	2
4.	1	1	0
5.	0	1	0

We first obtain the MLEs from `logistic` so that we can compare the estimates and associated statistics with the CMLEs from `exlogistic`.

```
. logistic hiv cd4 cd8, coef
Logistic regression                               Number of obs     =         47
                                                  LR chi2(2)        =      15.75
                                                  Prob > chi2       =     0.0004
Log likelihood = -20.751687                       Pseudo R2         =     0.2751

------------------------------------------------------------------------------
         hiv |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         cd4 |  -2.541669   .8392231    -3.03   0.002    -4.186517   -.8968223
         cd8 |   1.658586    .821113     2.02   0.043     .0492344    3.267938
       _cons |   .5132389   .6809007     0.75   0.451    -.8213019     1.84778
------------------------------------------------------------------------------

. exlogistic hiv cd4 cd8, coef
Enumerating sample-space combinations:
observation 1:  enumerations =          2
observation 2:  enumerations =          3
  (output omitted)
observation 46:  enumerations =        601
observation 47:  enumerations =        326
Exact logistic regression                         Number of obs =         47
                                                  Model score   =   13.34655
                                                  Pr >= score   =     0.0006

------------------------------------------------------------------------------
         hiv |      Coef.       Suff.  2*Pr(Suff.)     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         cd4 |  -2.387632          10      0.0004     -4.699633   -.8221807
         cd8 |   1.592366          12      0.0528     -.0137905    3.907876
------------------------------------------------------------------------------
```

`exlogistic` produced a log showing how many records are generated as it processes each observation. The primary purpose of the log is to provide feedback because generating the distribution can be time consuming, but we also see from the last entry that the joint distribution for the sufficient statistics for `cd4` and `cd8` conditioned on the total number of successes has 326 unique values (but a size of $\binom{47}{14} = 341{,}643{,}774{,}795$).

The statistics for `logistic` are based on asymptotics: for a large sample size, each Z statistic will be approximately normally distributed (with a mean of zero and a standard deviation of one) if the associated regression parameter is zero. The question is whether a sample size of 47 is large enough.

On the other hand, the p-values computed by `exlogistic` are from the conditional distributions of the sufficient statistics for each parameter given the sufficient statistics for all other parameters. In this sense, these p-values are exact. By default, `exlogistic` reports the sufficient statistics for the regression parameters and the probability of observing a more extreme value. These are single-parameter tests for H_0: $\beta_{\text{cd4}} = 0$ and H_0: $\beta_{\text{cd8}} = 0$ versus the two-sided alternatives. The conditional scores test, located in the coefficient table header, is testing that both H_0: $\beta_{\text{cd4}} = 0$ and H_0: $\beta_{\text{cd8}} = 0$. We find these p-values to be in fair agreement with the Wald and likelihood-ratio tests from `logistic`.

The confidence intervals for `exlogistic` are computed from the exact conditional distributions. The exact confidence intervals are asymmetrical about the estimate and are wider than the normal-based confidence intervals from `logistic`.

Both estimation techniques indicate that the incidence of HIV infection decreases with increasing CD4 blood serum levels and increases with increasing CD8 blood serum levels. The constant term is

missing from the exact logistic coefficient table because we conditioned out its observed sufficient statistic when tallying the joint distribution of the sufficient statistics for the cd4 and cd8 parameters.

The test() option provides two other test statistics used in exact logistic: the conditional scores test, test(score), and the conditional probabilities test, test(probability). For comparison, we display the individual parameter conditional scores tests.

```
. exlogistic, test(score) coef
Exact logistic regression                       Number of obs =         47
                                                Model score   =   13.34655
                                                Pr >= score   =     0.0006
```

hiv	Coef.	Score	Pr>=Score	[95% Conf.	Interval]
cd4	-2.387632	12.88022	0.0003	-4.699633	-.8221807
cd8	1.592366	4.604816	0.0410	-.0137905	3.907876

For the probabilities test, the probability statistic is computed from (1) in *Methods and formulas* with $\boldsymbol{\beta} = 0$. For this example, the p-value for the probabilities tests matches the scores tests so they are not displayed here.

◁

❑ Technical note

Typically, the value of θ, the constant term, is of little interest, as well as perhaps some of the parameters in $\boldsymbol{\beta}$, but we need to include all parameters in the model to correctly specify it. By conditioning out the nuisance parameters, we can reduce the size of the joint conditional distribution that is used to estimate the regression parameters of interest. The condvars() option allows you to specify a *varlist* of nuisance variables. By default, exlogistic conditions on the sufficient statistic of θ, which is the number of successes. You can save computation time and computer memory by using the condvars() option because infeasible values of the sufficient statistics associated with the variables in condvars() can be dropped from consideration before all n observations are processed.

Specifying some of your independent variables in condvars() will not change the estimated regression coefficients of the remaining independent variables. For instance, in example 1, if we instead type

```
. exlogistic hiv cd4, condvars(cd8) coef
```

the regression coefficient for cd4 (as well as all associated inference) will be identical.

One reason to have multiple variables in *indepvars* is to make conditional inference of several parameters simultaneously by using the terms() option. If you do not wish to test several parameters simultaneously, it may be more efficient to obtain estimates for individual variables by calling exlogistic multiple times with one variable in *indepvars* and all other variables listed in condvars(). The estimates will be the same as those with all variables in *indepvars*.

❑

❑ Technical note

If you fit a clogit (see [R] **clogit**) model to the HIV data from example 1, you will find that the estimates differ from those with exlogistic. (To fit the clogit model, you will have to create a group variable that includes all observations.) The regression estimates will be different because clogit conditions on the constant term only, whereas the estimates from exlogistic condition on the sufficient statistic of the other regression parameter as well as the constant term.

❑

▷ Example 2

The HIV data presented in table IV of Mehta and Patel (1995) are in a binomial form, where the variable `hiv` contains the HIV cases that tested positive and the variable `n` contains the number of individuals with the same CD4 and CD8 levels, the binomial number-of-trials parameter. Here *depvar* is `hiv`, and we use the `binomial(n)` option to identify the number-of-trials variable.

```
. use http://www.stata-press.com/data/r15/hiv_n
(prospective study of perinatal infection of HIV-1; binomial form)
. list
```

	cd4	cd8	hiv	n
1.	0	2	1	1
2.	1	2	2	2
3.	0	0	4	7
4.	1	1	4	12
5.	2	2	1	3
6.	1	0	2	7
7.	2	0	0	2
8.	2	1	0	13

Further, the `cd4` and `cd8` variables of the `hiv` dataset are actually factor variables, where each has the ordered levels of $(0, 1, 2)$. Another approach to the analysis is to use indicator variables, and following Mehta and Patel (1995), we used a 0–1 coding scheme that will give us the odds ratio of level 0 versus 2 and level 1 versus 2.

```
. generate byte cd4_0 = (cd4==0)
. generate byte cd4_1 = (cd4==1)
. generate byte cd8_0 = (cd8==0)
. generate byte cd8_1 = (cd8==1)
. exlogistic hiv cd4_0 cd4_1 cd8_0 cd8_1, terms(cd4=cd4_0 cd4_1,
> cd8=cd8_0 cd8_1) binomial(n) test(probability) saving(dist, replace) nolog
note: saving distribution to file dist.dta
note: CMLE estimate for cd4_0 is +inf; computing MUE
note: CMLE estimate for cd4_1 is +inf; computing MUE
note: CMLE estimate for cd8_0 is -inf; computing MUE
note: CMLE estimate for cd8_1 is -inf; computing MUE
Exact logistic regression                       Number of obs =         47
Binomial variable: n                            Model prob.   =   3.19e-06
                                                Pr <= prob.   =     0.0011
```

hiv	Odds Ratio	Prob.	Pr<=Prob.	[95% Conf.	Interval]
cd4		.0007183	0.0055		
cd4_0	18.82831*	.007238	0.0072	1.714079	+Inf
cd4_1	11.53732*	.0063701	0.0105	1.575285	+Inf
cd8		.0053212	0.0323		
cd8_0	.1056887*	.0289948	0.0290	0	1.072531
cd8_1	.0983388*	.0241503	0.0242	0	.9837203

```
(*) median unbiased estimates (MUE)
. matrix list e(sufficient)
e(sufficient)[1,4]
    cd4_0  cd4_1  cd8_0  cd8_1
r1      5      8      6      4
```

```
. display e(n_possible)
1091475
```

Here we used `terms()` to specify two terms in the model, `cd4` and `cd8`, that make up the `cd4` and `cd8` indicator variables. By doing so, we obtained a conditional probabilities test for `cd4`, simultaneously testing both `cd4_0` and `cd4_1`, and for `cd8`, simultaneously testing both `cd8_0` and `cd8_1`. The p-values for the two terms are 0.0055 and 0.0323, respectively.

This example also illustrates instances where the dependent variable is completely determined by the independent variables and CMLEs are infinite. If we try to obtain MLEs, `logistic` will drop each variable and then terminate with a no-data error, error number 2000.

```
. use http://www.stata-press.com/data/r15/hiv_n, clear
(prospective study of perinatal infection of HIV-1; binomial form)
. generate byte cd4_0 = (cd4==0)
. generate byte cd4_1 = (cd4==1)
. generate byte cd8_0 = (cd8==0)
. generate byte cd8_1 = (cd8==1)
. expand n
(39 observations created)
. logistic hiv cd4_0 cd4_1 cd8_0 cd8_1
note: cd4_0 != 0 predicts success perfectly
      cd4_0 dropped and 8 obs not used
note: cd4_1 != 0 predicts success perfectly
      cd4_1 dropped and 21 obs not used
note: cd8_0 != 0 predicts failure perfectly
      cd8_0 dropped and 2 obs not used
outcome = cd8_1 <= 0 predicts data perfectly
r(2000);
```

In example 2, `exlogistic` generated the joint conditional distribution of $T_{\text{cd4\_0}}$, $T_{\text{cd4\_1}}$, $T_{\text{cd8\_0}}$, and $T_{\text{cd8\_1}}$ given $M = 14$ (the number of individuals that tested positive), and for reference, we listed the observed sufficient statistics that are stored in the matrix `e(sufficient)`. Below we take that distribution and further condition on $T_{\text{cd4\_1}} = 8$, $T_{\text{cd8\_0}} = 6$, and $T_{\text{cd8\_1}} = 4$, giving the conditional distribution of $T_{\text{cd4\_0}}$. Here we see that the observed sufficient statistic $T_{\text{cd4\_0}} = 5$ is last in the sorted listing or, equivalently, $T_{\text{cd4\_0}}$ is at the domain boundary of the conditional probability distribution. When this occurs, the conditional probability distribution is monotonically increasing in $\beta_{\text{cd4\_0}}$ and a maximum does not exist.

```
. use dist, clear
. keep if cd4_1==8 & cd8_0==6 & cd8_1==4
(4,139 observations deleted)
. list, sep(0)
```

	_f_	cd4_0	cd4_1	cd8_0	cd8_1
1.	1668667	0	8	6	4
2.	18945542	1	8	6	4
3.	55801053	2	8	6	4
4.	55867350	3	8	6	4
5.	17423175	4	8	6	4
6.	1091475	5	8	6	4

When the CMLEs are infinite, the MUEs are computed (Hirji, Tsiatis, and Mehta 1989). For the cd4_0 estimate, we compute the value $\overline{\beta}_{\text{cd4\_0}}$ such that

$$\Pr(T_{\text{cd4\_0}} \geq 5 \mid \beta_{\text{cd4\_0}} = \overline{\beta}_{\text{cd4\_0}}, T_{\text{cd4\_1}} = 8, T_{\text{cd8\_0}} = 6, T_{\text{cd8\_1}} = 4, M = 14) = 1/2$$

using (1) in *Methods and formulas*.

The output is in agreement with example 1: there is an increase in risk of HIV infection for a CD4 blood serum level of 0 relative to a level of 2 and for a level of 1 relative to a level of 2; there is a decrease in risk of HIV infection for a CD8 blood serum level of 0 relative to a level of 2 and for a level of 1 relative to a level of 2.

We also displayed e(n_possible). This is the combinatorial coefficient associated with the observed sufficient statistics. The same value is found in the _f_ variable of the conditional distribution dataset listed above. The size of the distribution is $\binom{47}{14} = 341{,}643{,}774{,}795$. This can be verified by summing the _f_ variable of the generated conditional distribution dataset.

```
. use dist, clear
. summarize _f_, meanonly
. di %15.1f r(sum)
 341643774795.0
```

▷ Example 3

One can think of exact logistic regression as a covariate-adjusted exact binomial. To demonstrate this point, we will use exlogistic to compute a binomial confidence interval for m successes of n trials, by fitting the constant-only model, and we will compare it with the confidence interval computed by ci proportions (see [R] **ci**). We will use the saving() option to retain the dataset containing the feasible values for the constant term sufficient statistic, namely, the number of successes, m, given n trials and their associated combinatorial coefficients $\binom{n}{m}$, $m = 0, 1, \ldots, n$.

```
. input y
                y
  1. 1
  2. 0
  3. 1
  4. 0
  5. 1
  6. 1
  7. end
. ci proportions y
```

Variable	Obs	Proportion	Std. Err.	— Binomial Exact — [95% Conf.	Interval]
y	6	.6666667	.1924501	.2227781	.9567281

```
. exlogistic y, estconstant nolog coef saving(binom)
note: saving distribution to file binom.dta
Exact logistic regression
                                                Number of obs =          6
```

y	Coef.	Suff.	2*Pr(Suff.)	[95% Conf.	Interval]
_cons	.6931472	4	0.6875	-1.24955	3.096017

We use the postestimation program `estat predict` to transform the estimated constant term and its confidence bounds by using the inverse logit function, `invlogit()` (see [FN] **Mathematical functions**). The standard error for the estimated probability is computed using the delta method.

```
. estat predict
```

y	Predicted	Std. Err.	[95% Conf.	Interval]
Probability	0.6667	0.1925	0.2228	0.9567

```
. use binom, replace
. list, sep(0)
```

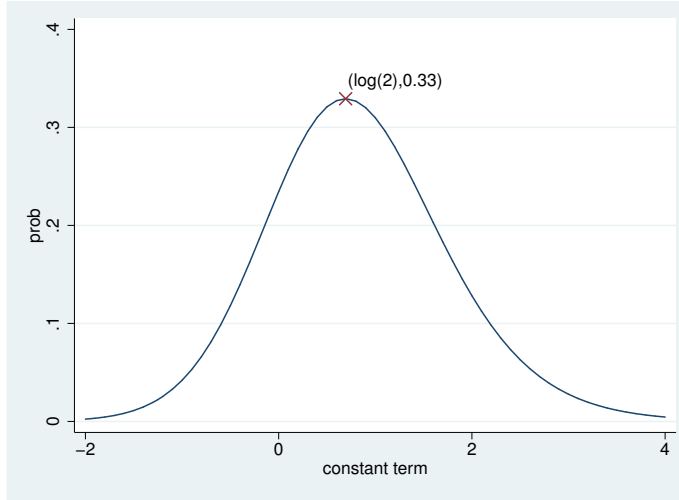
	_f_	_cons_
1.	1	0
2.	6	1
3.	15	2
4.	20	3
5.	15	4
6.	6	5
7.	1	6

Examining the listing of the generated data, the values contained in the variable `_cons_` are the feasible values of M, and the values contained in the variable `_f_` are the binomial coefficients $\binom{6}{m}$ with total $\sum_{m=0}^{6}\binom{6}{m} = 2^6 = 64$. In the coefficient table, the sufficient statistic for the constant term, labeled `Suff.`, is m = 4. This value is located at record 5 of the dataset. Therefore, the two-tailed probability of the sufficient statistic is computed as $0.6875 = 2(15 + 6 + 1)/64$.

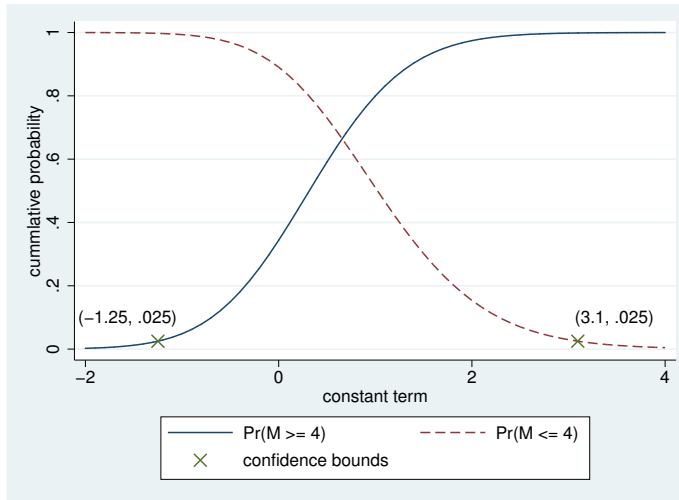
The constant term is the value of θ that maximizes the probability of observing $M = 4$; see (1) of *Methods and formulas*:

$$\Pr(M = 4|\theta) = \frac{15e^{4\alpha}}{1 + 6e^{\alpha} + 15e^{2\alpha} + 20e^{3\alpha} + 15e^{4\alpha} + 6e^{5\alpha} + e^{6\alpha}}$$

The maximum is at the value $\theta = \log 2$, which is demonstrated in the figure below.



The lower and upper confidence bounds are the values of θ such that $\Pr(M \geq 4|\theta) = 0.025$ and $\Pr(M \leq 4|\theta) = 0.025$, respectively. These probabilities are plotted in the figure below for $\theta \in [-2, 4]$.



Example 4

This example demonstrates the `group()` option, which allows the analysis of stratified data. Here the logistic model is

$$\log\left(\frac{\pi_{ik}}{1-\pi_{ik}}\right) = \theta_k + \mathbf{x}_{ki}\boldsymbol{\beta}$$

where k indexes the s strata, $k = 1, \ldots, s$, and θ_k is the strata-specific constant term whose sufficient statistic is $M_k = \sum_{i=1}^{n_k} Y_{ki}$.

Mehta and Patel (1995) use a case–control study to demonstrate this model, which is useful in comparing the estimates from `exlogistic` and `clogit`. This study was intended to determine the role of birth complications in people with schizophrenia (Garsd 1988). Siblings from seven families took part in the study, and each individual was classified as normal or schizophrenic. A birth complication index is recorded for each individual that ranges from 0, an uncomplicated birth, to 15, a very complicated birth. Some of the frequencies contained in variable `f` are greater than 1, and these count different births at different times where the individual has the same birth complications index, found in variable `BCindex`.

```
. use http://www.stata-press.com/data/r15/schizophrenia, clear
(case-control study on birth complications for people with schizophrenia)
. list, sepby(family)
```

	family	BCindex	schizo	f
1.	1	6	0	1
2.	1	7	0	1
3.	1	3	0	2
4.	1	2	0	3
5.	1	5	0	1
6.	1	0	0	1
7.	1	15	1	1
8.	2	2	1	1
9.	2	0	0	1
10.	3	2	0	1
11.	3	9	1	1
12.	3	1	0	1
13.	4	2	1	1
14.	4	0	0	4
15.	5	3	1	1
16.	5	6	0	1
17.	5	0	1	1
18.	6	3	0	1
19.	6	0	1	1
20.	6	0	0	2
21.	7	2	0	1
22.	7	6	1	1

```
. exlogistic schizo BCindex [fw=f], group(family) test(score) coef

Enumerating sample-space combinations:
observation 1:   enumerations =          2
observation 2:   enumerations =          3
observation 3:   enumerations =          4
observation 4:   enumerations =          5
observation 5:   enumerations =          6
observation 6:   enumerations =          7
  (output omitted)
observation 21:  enumerations =         72
observation 22:  enumerations =         40

Exact logistic regression                       Number of obs     =         29
Group variable: family                          Number of groups  =          7

                                                Obs per group:
                                                              min =          2
                                                              avg =        4.1
                                                              max =         10

                                                Model score       =   6.328033
                                                Pr >= score       =     0.0167

------------------------------------------------------------------------------
      schizo |      Coef.       Score    Pr>=Score     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     BCindex |    .3251178    6.328033      0.0167       .0223423    .7408832
------------------------------------------------------------------------------
```

The asymptotic alternative for this model can be estimated using `clogit` (equivalently, `xtlogit, fe`) and is listed below for comparison. We must expand the data because `clogit` will not accept frequency weights if they are not constant within the groups.

```
. expand f
(7 observations created)

. clogit schizo BCindex, group(family) nolog
note: multiple positive outcomes within groups encountered.

Conditional (fixed-effects) logistic regression

                                                Number of obs     =         29
                                                LR chi2(1)        =       5.20
                                                Prob > chi2       =     0.0226
Log likelihood = -6.2819819                     Pseudo R2         =     0.2927

------------------------------------------------------------------------------
      schizo |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     BCindex |    .3251178   .1678981     1.94   0.053    -.0039565     .654192
------------------------------------------------------------------------------
```

Both techniques compute the same regression estimate for the `BCindex`, which might not be too surprising because both estimation techniques condition on the total number of successes in each group. The difference lies in the p-values and confidence intervals. The p-value testing $H_0\colon \beta_{\text{BCindex}} = 0$ is approximately 0.0167 for the exact conditional scores test and 0.053 for the asymptotic Wald test. Moreover, the exact confidence interval is asymmetric about the estimate and does not contain zero.

◁

❑ Technical note

The `memory(#)` option limits the amount of memory that `exlogistic` will consume when computing the conditional distribution of the parameter sufficient statistics. `memory()` is independent of the data maximum memory setting (see `set max_memory` in [D] **memory**), and it is possible

for exlogistic to exceed the memory limit specified in set max_memory without terminating. By default, a log is provided that displays the number of enumerations (the size of the conditional distribution) after processing each observation. Typically, you will see the number of enumerations increase, and then at some point they will decrease as the multivariate shift algorithm (Hirji, Mehta, and Patel 1987) determines that some of the enumerations cannot achieve the observed sufficient statistics of the conditioning variables. When the algorithm is complete, however, it is necessary to store the conditional distribution of the parameter sufficient statistics as a dataset. It is possible, therefore, to get a memory error when the algorithm has completed if there is not enough memory to store the conditional distribution.

❑

❑ Technical note

Computing the conditional distributions and reported statistics requires data sorting and numerical comparisons. If there is at least one single-precision variable specified in the model, exlogistic will make comparisons with a relative precision of 2^{-5}. Otherwise, a relative precision of 2^{-11} is used. Be careful if you use recast to promote a single-precision variable to double precision (see [D] **recast**). You might try listing the data in full precision (maybe %20.15g; see [D] **format**) to make sure that this is really what you want. See [D] **data types** for information on precision of numeric storage types.

❑

Stored results

exlogistic stores the following in e():

Scalars

e(N)	number of observations
e(k_groups)	number of groups
e(n_possible)	number of distinct possible outcomes where sum(sufficient) equals observed e(sufficient)
e(n_trials)	binomial number-of-trials parameter
e(sum_y)	sum of *depvar*
e(k_indvars)	number of independent variables
e(k_terms)	number of model terms
e(k_condvars)	number of conditioning variables
e(condcons)	conditioned on the constant(s) indicator
e(midp)	mid-p-value rule indicator
e(eps)	relative difference tolerance

Macros

e(cmd)	exlogistic
e(cmdline)	command as typed
e(title)	title in estimation output
e(depvar)	name of dependent variable
e(indvars)	independent variables
e(condvars)	conditional variables
e(groupvar)	group variable
e(binomial)	binomial number-of-trials variable
e(terms)	term names set in option terms()
e(level)	confidence level
e(wtype)	weight type
e(wexp)	weight expression
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b
e(estat_cmd)	program used to implement estat
e(marginsnotok)	predictions disallowed by margins

Matrices

e(b)	coefficient vector
e(mue_indicators)	indicator for elements of e(b) estimated using MUE instead of CMLE
e(se)	e(b) standard errors (CMLEs only)
e(ci)	matrix of e(level) confidence intervals for e(b)
e(sum_y_groups)	sum of e(depvar) for each group
e(N_g)	number of observations in each group
e(sufficient)	sufficient statistics for e(b)
e(p_sufficient)	p-value for e(sufficient)
e(scoretest)	conditional scores tests for *indepvars*
e(p_scoretest)	p-values for e(scoretest)
e(probtest)	conditional probabilities tests for *indepvars*
e(p_probtest)	p-value for e(probtest)
e(scoretest_m)	conditional scores tests for model terms
e(p_scoretest_m)	p-value for e(scoretest_m)
e(probtest_m)	conditional probabilities tests for model terms
e(p_probtest_m)	p-value for e(probtest_m)

Functions

e(sample)	marks estimation sample

Methods and formulas

Methods and formulas are presented under the following headings:

Sufficient statistics
Conditional distribution and CMLE
Median unbiased estimates and exact CI
Conditional hypothesis tests
Sufficient-statistic p-value

Sufficient statistics

Let $\{Y_1, Y_2, \ldots, Y_n\}$ be a set of n independent Bernoulli random variables, each of which can realize two outcomes, $\{0, 1\}$. For each $i = 1, \ldots, n$, we observe $Y_i = y_i$, and associated with each observation is the covariate row vector of length p, $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$. Denote $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ to be the column vector of regression parameters and θ to be the constant. The sufficient statistic for β_j is $T_j = \sum_{i=1}^n Y_i x_{ij}$, $j = 1, \ldots, p$, and for θ is $M = \sum_{i=1}^n Y_i$. We observe $T_j = t_j$, $t_j = \sum_{i=1}^n y_i x_{ij}$, and $M = m$, $m = \sum_{i=1}^n y_i$. The probability of observing $(Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n)$ is

$$\Pr(Y_1 = y_1, \ldots, Y_n = y_n \mid \boldsymbol{\beta}, \mathbf{X}) = \frac{\exp(m\theta + \mathbf{t}\boldsymbol{\beta})}{\prod_{i=1}^n \{1 + \exp(\theta + \mathbf{x}_i\boldsymbol{\beta})\}}$$

where $\mathbf{t} = (t_1, \ldots, t_p)$ and $\mathbf{X} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T)^T$.

The joint distribution of the sufficient statistics $\mathbf{T}$ is obtained by summing over all possible binary sequences $Y_1, \ldots, Y_n$ such that $\mathbf{T} = \mathbf{t}$ and $M = m$. This probability function is

$$\Pr(T_1 = t_1, \ldots, T_p = t_p, M = m \mid \boldsymbol{\beta}, \mathbf{X}) = \frac{c(\mathbf{t}, m)\exp(m\theta + \mathbf{t}\boldsymbol{\beta})}{\prod_{i=1}^n \{1 + \exp(\theta + \mathbf{x}_i\boldsymbol{\beta})\}}$$

where $c(\mathbf{t}, m)$ is the combinatorial coefficient of $(\mathbf{t}, m)$ or the number of distinct binary sequences $Y_1, \ldots, Y_n$ such that $\mathbf{T} = \mathbf{t}$ and $M = m$ (Cox and Snell 1989).

Conditional distribution and CMLE

Without loss of generality, we will restrict our discussion to computing the CMLE of β_1. If we condition on observing $M = m$ and $T_2 = t_2, \ldots, T_p = t_p$, the probability function of $(T_1 \mid \beta_1, T_2 = t_2, \ldots, T_p = t_p, M = m)$ is

$$\Pr(T_1 = t_1 \mid \beta_1, T_2 = t_2, \ldots, T_p = t_p, M = m) = \frac{c(\mathbf{t}, m)e^{t_1\beta_1}}{\sum_u c(u, t_2, \ldots, t_p, m)e^{u\beta_1}} \tag{1}$$

where the sum in the denominator is over all possible values of T_1 such that $M = m$ and $T_2 = t_2, \ldots, T_p = t_p$ and $c(u, t_2, \ldots, t_p, m)$ is the combinatorial coefficient of $(u, t_2, \ldots, t_p, m)$ (Cox and Snell 1989). The CMLE for β_1 is the value $\widehat{\beta}_1$ that maximizes the log of (1). This optimization task is carried out by `ml`, using the conditional frequency distribution of $(T_1 \mid T_2 = t_2, \ldots, T_p = t_p, M = m)$ as a dataset. Generating the joint conditional distribution is efficiently computed using the multivariate shift algorithm described by Hirji, Mehta, and Patel (1987).

Difficulties in computing $\widehat{\beta}_1$ arise if the observed $(T_1 = t_1, \ldots, T_p = t_p, M = m)$ lies on the boundaries of the distribution of $(T_1 \mid T_2 = t_2, \ldots, T_p = t_p, M = m)$, where the conditional probability function is monotonically increasing (or decreasing) in β_1. Here the CMLE is plus infinity if it is on the upper boundary, $\Pr(T_1 \leq t_1 | T_2 = t_2, \ldots, T_p = t_p, M = m) = 1$, and is minus infinity if it is on the lower boundary of the distribution, $\Pr(T_1 \geq t_1 | T_2 = t_2, \ldots, T_p = t_p, M = m) = 1$. This concept is demonstrated in example 2. When infinite CMLEs occur, the MUE is computed.

Median unbiased estimates and exact CI

The MUE is computed using the technique outlined by Hirji, Tsiatis, and Mehta (1989). First, we find the values of $\beta_1^{(u)}$ and $\beta_1^{(l)}$ such that

$$\begin{aligned}&\Pr(T_1 \leq t_1 \mid \beta_1 = \beta_1^{(u)}, T_2 = t_2, \ldots, T_p = t_p, M = m) = \\ &\Pr(T_1 \geq t_1 \mid \beta_1 = \beta_1^{(l)}, T_2 = t_2, \ldots, T_p = t_p, M = m) = 1/2\end{aligned} \tag{2}$$

The MUE is then $\overline{\beta}_1 = \left(\beta_1^{(l)} + \beta_1^{(u)}\right)/2$. However, if T_1 is equal to the minimum of the domain of the conditional distribution, $\beta^{(l)}$ does not exist and $\overline{\beta}_1 = \beta^{(u)}$. If T_1 is equal to the maximum of the domain of the conditional distribution, $\beta^{(u)}$ does not exist and $\overline{\beta}_1 = \beta^{(l)}$.

Confidence bounds for β are computed similarly, except that we substitute $\alpha/2$ for $1/2$ in (2), where $1 - \alpha$ is the confidence level. Here $\beta_1^{(l)}$ would then be the lower confidence bound and $\beta_1^{(u)}$ would be the upper confidence bound (see example 3).

Conditional hypothesis tests

To test H_0: $\beta_1 = 0$ versus H_1: $\beta_1 \neq 0$, we obtain the exact p-value from $\sum_{u \in E} f_1(u) - f_1(t_1)/2$ if the mid-p-value rule is used and $\sum_{u \in E} f_1(u)$ otherwise. Here E is a critical region, and we define $f_1(u) = \Pr(T_1 = u \mid \beta_1 = 0, T_2 = t_2, \ldots, T_p = t_p, M = m)$ for ease of notation. There are two popular ways to define the critical region: the conditional probabilities test and the conditional scores test (Mehta and Patel 1995). The critical region when using the conditional probabilities test is all values of the sufficient statistic for β_1 that have a probability less than or equal to that of the observed t_1, $E_p = \{u : \ f_1(u) \leq f_1(t_1)\}$. The critical region of the conditional scores test is defined as all values of the sufficient statistic for β_1 such that its score is greater than or equal to that of t_1,

$$E_s = \left\{u : \ (u - \mu_1)^2/\sigma_1^2 \geq (t_1 - \mu_1)^2/\sigma_1^2)\right\}$$

Here μ_1 and σ_1^2 are the mean and variance of $(T_1 \mid \beta_1 = 0, T_2 = t_2, \ldots, T_p = t_p, M = m)$.

The score statistic is defined as

$$\left\{\frac{\partial \ell(\beta)}{\partial \beta}\right\}^2 \left[-E\left\{\frac{\partial^2 \ell(\beta)}{\partial \beta^2}\right\}\right]^{-1}$$

evaluated at H_0: $\beta = 0$, where ℓ is the log of (1). The score test simplifies to $(t - E[T|\beta])^2/\mathrm{var}(T|\beta)$ (Hirji 2006), where the mean and variance are computed from the conditional distribution of the sufficient statistic with $\beta = 0$ and t is the observed sufficient statistic.

Sufficient-statistic p-value

The p-value for testing $H_0: \beta_1 = 0$ versus the two-sided alternative when $(T_1 = t_1 | T_2 = t_2, \ldots, T_p = t_p)$ is computed as $2 \times \min(p_l, p_u)$, where

$$p_l = \frac{\sum_{u \le t_1} c(u, t_2, \ldots, t_p, m)}{\sum_u c(u, t_2, \ldots, t_p, m)}$$

$$p_u = \frac{\sum_{u \ge t_1} c(u, t_2, \ldots, t_p, m)}{\sum_u c(u, t_2, \ldots, t_p, m)}$$

It is the probability of observing a more extreme T_1.

References

Cox, D. R., and E. J. Snell. 1989. *Analysis of Binary Data*. 2nd ed. London: Chapman & Hall.

Garsd, A. 1988. Schizophrenia and birth complications. Unpublished manuscript.

Hirji, K. F. 2006. *Exact Analysis of Discrete Data*. Boca Raton: Chapman & Hall/CRC.

Hirji, K. F., C. R. Mehta, and N. R. Patel. 1987. Computing distributions for exact logistic regression. *Journal of the American Statistical Association* 82: 1110–1117.

Hirji, K. F., A. A. Tsiatis, and C. R. Mehta. 1989. Median unbiased estimation for binary data. *American Statistician* 43: 7–11.

Hutto, C., W. P. Parks, S. Lai, M. T. Mastrucci, C. Mitchell, J. Muñoz, E. Trapido, I. M. Master, and G. B. Scott. 1991. A hospital-based prospective study of perinatal infection with human immunodeficiency virus type 1. *Journal of Pediatrics* 118: 347–353.

Mehta, C. R., and N. R. Patel. 1995. Exact logistic regression: Theory and examples. *Statistics in Medicine* 14: 2143–2160.

Also see

[R] **exlogistic postestimation** — Postestimation tools for exlogistic

[R] **binreg** — Generalized linear models: Extensions to the binomial family

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

[R] **expoisson** — Exact Poisson regression

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

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

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