binreg — Generalized linear models: Extensions to the binomial family

Description

`binreg` fits generalized linear models for the binomial family. It estimates odds ratios, risk ratios, health ratios, and risk differences. The available links are

Option	Implied link	Parameter
`or`	`logit`	odds ratios = exp(β)
`rr`	`log`	risk ratios = exp(β)
`hr`	`log complement`	health ratios = exp(β)
`rd`	`identity`	risk differences = β

Estimates of odds, risk, and health ratios are obtained by exponentiating the appropriate coefficients. The `or` option produces the same results as Stata's `logistic` command, and `or coefficients` yields the same results as the `logit` command. When no link is specified, `or` is assumed.

Quick start

Report odds ratios from a model of `y` on `x1` and `x2` using a logit link

```
binreg y x1 x2, or
```

Use the log link and report risk ratios

```
binreg y x1 x2, rr
```

Use the identity link and report risk differences

```
binreg y x1 x2, rd
```

Same as above, but with data stored as the number of successes, `ys`, out of `n` trials

```
binreg ys x1 x2, rd n(n)
```

Menu

Statistics > Generalized linear models > GLM for the binomial family

Syntax

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

options	Description
Model	
`noconstant`	suppress constant term
`or`	use logit link and report odds ratios
`rr`	use log link and report risk ratios
`hr`	use log-complement link and report health ratios
`rd`	use identity link and report risk differences
`n(`#|*varname*`)`	use # or *varname* for number of trials
`exposure(`*varname*`)`	include ln(*varname*) in model with coefficient constrained to 1
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
`mu(`*varname*`)`	use *varname* as the initial estimate for the mean of *depvar*
`init(`*varname*`)`	synonym for `mu(`*varname*`)`
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `eim`, `robust`, `cluster` *clustvar*, `oim`, `opg`, `bootstrap`, `jackknife`, `hac` *kernel*, `jackknife1`, or `unbiased`
`t(`*varname*`)`	variable name corresponding to time
`vfactor(`#`)`	multiply variance matrix by scalar #
`disp(`#`)`	quasilikelihood multiplier
`scale(x2`|`dev`|#`)`	set the scale parameter; default is `scale(1)`
Reporting	
`level(`#`)`	set confidence level; default is `level(95)`
`coefficients`	report nonexponentiated coefficients
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
`irls`	use iterated, reweighted least-squares optimization; the default
`ml`	use maximum likelihood optimization
maximize_options	control the maximization process; seldom used
`fisher(`#`)`	Fisher scoring steps
`search`	search for good starting values
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

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

depvar and *indepvars* may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`bayes`, `bayesboot`, `bootstrap`, `by`, `collect`, `fp`, `jackknife`, `mi estimate`, `rolling`, and `statsby` are allowed; see **[U] 11.1.10 Prefix commands**. For more details, see [BAYES] **bayes: binreg**.

`vce(bootstrap)`, `vce(jackknife)`, and `vce(jackknife1)` are not allowed with the `mi estimate` prefix; see [MI] **mi estimate**.

Weights are not allowed with the `bootstrap` prefix; see [R] **bootstrap**.

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

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

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

Options

Model

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

`or` requests the logit link and results in odds ratios if `coefficients` is not specified.

`rr` requests the log link and results in risk ratios if `coefficients` is not specified.

`hr` requests the log-complement link and results in health ratios if `coefficients` is not specified.

`rd` requests the identity link and results in risk differences.

`n(`#|*varname*`)` specifies either a constant integer to use as the denominator for the binomial family or a variable that holds the denominator for each observation.

`exposure(`*varname*`)`, `offset(`*varname*`)`, `constraints(`*constraints*`)`; see [R] **Estimation options**. `constraints(`*constraints*`)` is not allowed with `irls`.

`mu(`*varname*`)` specifies *varname* containing an initial estimate for the mean of *depvar*. This option can be useful if you encounter convergence difficulties. `init(`*varname*`)` is a synonym.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster` *clustvar*), that are derived from asymptotic theory (`oim`, `opg`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [R] ***vce_option***.

`vce(eim)`, the default, uses the expected information matrix (EIM) for the variance estimator.

`binreg` also allows the following:

`vce(hac` *kernel* [#]`)` specifies that a heteroskedasticity- and autocorrelation-consistent (HAC) variance estimate be used. HAC refers to the general form for combining weighted matrices to form the variance estimate. There are three kernels built into `binreg`. *kernel* is a user-written program or one of

`nwest` | `gallant` | `anderson`

If # is not specified, $N-2$ is assumed.

`vce(jackknife1)` specifies that the one-step jackknife estimate of variance be used.

`vce(unbiased)` specifies that the unbiased sandwich estimate of variance be used.

`t(`*varname*`)` specifies the variable name corresponding to time; see [TS] **tsset**. `binreg` does not always need to know `t()`, though it does if `vce(hac` ...`)` is specified. Then you can either specify the time variable with `t()`, or you can `tsset` your data before calling `binreg`. When the time variable is required, `binreg` assumes that the observations are spaced equally over time.

`vfactor(#)` specifies a scalar by which to multiply the resulting variance matrix. This option allows users to match output with other packages, which may apply degrees of freedom or other small-sample corrections to estimates of variance.

`disp(#)` multiplies the variance of *depvar* by # and divides the deviance by #. The resulting distributions are members of the quasilikelihood family. This option is not allowed with option `ml`.

`scale(x2`|`dev`|`#)` overrides the default scale parameter. This option is allowed only with Hessian (information matrix) variance estimates.

By default, `scale(1)` is assumed for the discrete distributions (binomial, Poisson, and negative binomial), and `scale(x2)` is assumed for the continuous distributions (Gaussian, gamma, and inverse Gaussian).

`scale(x2)` specifies that the scale parameter be set to the Pearson χ^2 (or generalized χ^2) statistic divided by the residual degrees of freedom, which is recommended by McCullagh and Nelder (1989) as a good general choice for continuous distributions.

`scale(dev)` sets the scale parameter to the deviance divided by the residual degrees of freedom. This option provides an alternative to `scale(x2)` for continuous distributions and overdispersed or underdispersed discrete distributions. This option is not allowed with option `ml`.

`scale(#)` sets the scale parameter to #.

Reporting

`level(#)`, `noconstant`; see [R] **Estimation options**.

`coefficients` displays the nonexponentiated coefficients and corresponding standard errors and confidence intervals. This option has no effect when the `rd` option is specified, because it always presents the nonexponentiated coefficients.

`nocnsreport`; see [R] **Estimation options**.

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

Maximization

`irls` requests iterated, reweighted least-squares (IRLS) optimization of the deviance instead of Newton–Raphson optimization of the log likelihood. This option is the default.

`ml` requests that optimization be carried out by using Stata's `ml` command; see [R] **ml**.

maximize_options: `difficult`, `technique(`*algorithm_spec*`)`, `iterate(#)`, [`no`]`log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(`*init_specs*`)`; see [R] **Maximize**. These options are seldom used.

Setting the optimization method to `ml`, with `technique()` set to something other than BHHH, changes the *vcetype* to `vce(oim)`. Specifying `technique(bhhh)` changes *vcetype* to `vce(opg)`.

Unless option `ml` is specified, only *maximize_options* `iterate()`, `nolog`, `trace`, and `ltolerance()` are allowed. With IRLS optimization, the convergence criterion is satisfied when the absolute change in deviance from one iteration to the next is less than or equal to `ltolerance()`, where `ltolerance(1e-6)` is the default.

`fisher(#)` specifies the number of Newton–Raphson steps that should use the Fisher scoring Hessian or EIM before switching to the observed information matrix (OIM). This option is available only if `ml` is specified and is useful only for Newton–Raphson optimization.

`search` specifies that the command search for good starting values. This option is available only if `ml` is specified and is useful only for Newton–Raphson optimization.

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

`collinear`, `coeflegend`; see [R] **Estimation options**. `collinear` is not allowed with `irls`.

Remarks and examples

Wacholder (1986) suggests methods for estimating risk ratios and risk differences from prospective binomial data. These estimates are obtained by selecting the proper link functions in the generalized linear-model framework. (See *Methods and formulas* for details; also see [R] **glm**.)

▷ Example 1

Wacholder (1986) presents an example, using data from Wright et al. (1983), of an investigation of the relationship between alcohol consumption and the risk of a low-birthweight baby. Covariates examined included whether the mother smoked (yes or no), mother's social class (three levels), and drinking frequency (light, moderate, or heavy). The data for the 18 possible categories determined by the covariates are illustrated below.

Let's first describe the data and list a few observations.

```
. use https://www.stata-press.com/data/r19/binreg
(Alcohol and low-birthweight baby)

. list

     +---------------------------------------------------------------+
     | category   n_lbw_~s   n_women    alcohol      smokes   social |
     |---------------------------------------------------------------|
  1. |        1         11        84      Heavy   Nonsmoker        1 |
  2. |        2          5        79   Moderate   Nonsmoker        1 |
  3. |        3         11       169      Light   Nonsmoker        1 |
  4. |        4          6        28      Heavy      Smoker        1 |
  5. |        5          3        13   Moderate      Smoker        1 |
     |---------------------------------------------------------------|
  6. |        6          1        26      Light      Smoker        1 |
  7. |        7          4        22      Heavy   Nonsmoker        2 |
  8. |        8          3        25   Moderate   Nonsmoker        2 |
  9. |        9         12       162      Light   Nonsmoker        2 |
 10. |       10          4        17      Heavy      Smoker        2 |
     |---------------------------------------------------------------|
 11. |       11          2         7   Moderate      Smoker        2 |
 12. |       12          6        38      Light      Smoker        2 |
 13. |       13          0        14      Heavy   Nonsmoker        3 |
 14. |       14          1        18   Moderate   Nonsmoker        3 |
 15. |       15         12        91      Light   Nonsmoker        3 |
     |---------------------------------------------------------------|
 16. |       16          7        19      Heavy      Smoker        3 |
 17. |       17          2        18   Moderate      Smoker        3 |
 18. |       18          8        70      Light      Smoker        3 |
     +---------------------------------------------------------------+
```

Each observation corresponds to one of the 18 covariate structures. The number of low-birthweight babies from `n_women` in each category is given by the `n_lbw_babies` variable.

We begin by estimating risk ratios:

```
. binreg n_lbw_babies i.soc i.alc i.smo, n(n_women) rr
Iteration 1:  Deviance =   14.2879
Iteration 2:  Deviance =    13.607
Iteration 3:  Deviance =  13.60503
Iteration 4:  Deviance =  13.60503

Generalized linear models                          Number of obs    =         18
Optimization     : MQL Fisher scoring              Residual df      =         12
                   (IRLS EIM)                      Scale parameter =           1
Deviance         =   13.6050268                    (1/df) Deviance =    1.133752
Pearson          =  11.51517095                    (1/df) Pearson  =    .9595976

Variance function: V(u) = u*(1-u/n_women)          [Binomial]
Link function    : g(u) = ln(u/n_women)            [Log]

                                                   BIC              =  -21.07943

------------------------------------------------------------------------------
             |                 EIM
n_lbw_babies | Risk ratio   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      social |
          2  |   1.340001   .3127382     1.25   0.210      .848098     2.11721
          3  |   1.349487   .3291488     1.23   0.219     .8366715    2.176619
             |
     alcohol |
    Moderate |   1.191157   .3265354     0.64   0.523     .6960276    2.038503
       Heavy |   1.974078   .4261751     3.15   0.002     1.293011    3.013884
             |
      smokes |
      Smoker |   1.648444    .332875     2.48   0.013     1.109657    2.448836
       _cons |   .0630341   .0128061   -13.61   0.000     .0423297    .0938656
------------------------------------------------------------------------------
Note: _cons estimates baseline risk.
```

By default, Stata reports the risk ratios (the exponentiated regression coefficients) estimated by the model. We can see that the risk ratio comparing heavy drinkers with light drinkers, after adjusting for smoking and social class, is 1.974078. That is, mothers who drink heavily during their pregnancy have approximately twice the risk of delivering low-birthweight babies as mothers who are light drinkers.

The nonexponentiated coefficients can be obtained with the `coefficients` option:

```
. binreg n_lbw_babies i.soc i.alc i.smo, n(n_women) rr coefficients
Iteration 1:  Deviance =   14.2879
Iteration 2:  Deviance =    13.607
Iteration 3:  Deviance =  13.60503
Iteration 4:  Deviance =  13.60503

Generalized linear models                          Number of obs   =         18
Optimization     : MQL Fisher scoring              Residual df     =         12
                   (IRLS EIM)                      Scale parameter =          1
Deviance         =   13.6050268                    (1/df) Deviance =   1.133752
Pearson          =  11.51517095                    (1/df) Pearson  =   .9595976

Variance function: V(u) = u*(1-u/n_women)          [Binomial]
Link function    : g(u) = ln(u/n_women)            [Log]

                                                   BIC             =  -21.07943

------------------------------------------------------------------------------
             |                 EIM
n_lbw_babies | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      social |
          2  |   .2926702   .2333866     1.25   0.210    -.1647591    .7500994
          3  |   .2997244   .2439066     1.23   0.219    -.1783238    .7777726
             |
     alcohol |
   Moderate  |   .1749248    .274133     0.64   0.523     -.362366    .7122156
      Heavy  |   .6801017   .2158856     3.15   0.002     .2569737     1.10323
             |
      smokes |
     Smoker  |   .4998317   .2019329     2.48   0.013     .1040505    .8956129
       _cons |  -2.764079   .2031606   -13.61   0.000    -3.162266   -2.365891
------------------------------------------------------------------------------
```

Risk differences are obtained with the `rd` option:

```
. binreg n_lbw_babies i.soc i.alc i.smo, n(n_women) rd
Iteration 1:  Deviance =  18.67277
Iteration 2:  Deviance =  14.94364
Iteration 3:  Deviance =   14.9185
Iteration 4:  Deviance =  14.91762
Iteration 5:  Deviance =  14.91758
Iteration 6:  Deviance =  14.91758
Iteration 7:  Deviance =  14.91758

Generalized linear models                         Number of obs   =         18
Optimization     : MQL Fisher scoring             Residual df     =         12
                   (IRLS EIM)                     Scale parameter =          1
Deviance         =  14.91758277                   (1/df) Deviance =   1.243132
Pearson          =  12.60353235                   (1/df) Pearson  =   1.050294

Variance function: V(u) = u*(1-u/n_women)         [Binomial]
Link function    : g(u) = u/n_women               [Identity]

                                                  BIC             =  -19.76688

------------------------------------------------------------------------------
             |                 EIM
n_lbw_babies | Risk diff.   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      social |
          2  |   .0263817   .0232124     1.14   0.256    -.0191137    .0718771
          3  |   .0365553   .0268668     1.36   0.174    -.0161026    .0892132
             |
     alcohol |
   Moderate  |   .0122539   .0257713     0.48   0.634    -.0382569    .0627647
      Heavy  |   .0801291   .0302878     2.65   0.008      .020766    .1394921
             |
      smokes |
     Smoker  |   .0542415   .0270838     2.00   0.045     .0011582    .1073248
       _cons |    .059028   .0160693     3.67   0.000     .0275327    .0905232
------------------------------------------------------------------------------
```

The risk difference between heavy drinkers and light drinkers is 0.0801291. Because the risk differences are obtained directly from the coefficients estimated by using the identity link, the `coefficients` option would have no effect here.

Health ratios are obtained with the `hr` option. The health ratios (exponentiated coefficients for the log-complement link) are reported directly.

```
. binreg n_lbw_babies i.soc i.alc i.smo, n(n_women) hr
Iteration 1:  Deviance =  21.15233
Iteration 2:  Deviance =  15.16467
Iteration 3:  Deviance =  15.13205
Iteration 4:  Deviance =  15.13114
Iteration 5:  Deviance =  15.13111
Iteration 6:  Deviance =  15.13111
Iteration 7:  Deviance =  15.13111
Generalized linear models                         Number of obs    =         18
Optimization     : MQL Fisher scoring             Residual df      =         12
                   (IRLS EIM)                     Scale parameter =          1
Deviance         =  15.13110545                   (1/df) Deviance =   1.260925
Pearson          =  12.84203917                   (1/df) Pearson  =    1.07017
Variance function: V(u) = u*(1-u/n_women)         [Binomial]
Link function    : g(u) = ln(1-u/n_women)         [Log complement]
                                                  BIC              =  -19.55336

------------------------------------------------------------------------------
             |                 EIM
n_lbw_babies | Hlth ratio   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      social |
          2  |   .9720541    .024858    -1.11   0.268     .9245342    1.022017
          3  |   .9597182   .0290412    -1.36   0.174     .9044535     1.01836
             |
     alcohol |
   Moderate  |   .9871517   .0278852    -0.46   0.647     .9339831    1.043347
      Heavy  |   .9134243   .0325726    -2.54   0.011     .8517631    .9795493
             |
      smokes |
     Smoker  |   .9409983   .0296125    -1.93   0.053     .8847125    1.000865
       _cons |   .9409945   .0163084    -3.51   0.000     .9095674    .9735075
------------------------------------------------------------------------------
Note: _cons estimates baseline health (probability of no disease).
```

To see the nonexponentiated coefficients, we could specify the `coefficients` option.

◁

Stored results

`binreg, irls` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(k)`	number of parameters
`e(k_eq_model)`	number of equations in overall model test
`e(df_m)`	model degrees of freedom
`e(df)`	residual degrees of freedom
`e(phi)`	model scale parameter
`e(disp)`	dispersion parameter
`e(bic)`	model BIC
`e(N_clust)`	number of clusters
`e(deviance)`	deviance
`e(deviance_s)`	scaled deviance
`e(deviance_p)`	Pearson deviance
`e(deviance_ps)`	scaled Pearson deviance
`e(dispers)`	dispersion
`e(dispers_s)`	scaled dispersion
`e(dispers_p)`	Pearson dispersion
`e(dispers_ps)`	scaled Pearson dispersion
`e(vf)`	factor set by `vfactor()`, `1` if not set
`e(rank)`	rank of `e(V)`
`e(rc)`	return code

Macros

`e(cmd)`	`binreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(eform)`	`eform()` option implied by `or`, `rr`, `hr`, or `rd`
`e(varfunc)`	program to calculate variance function
`e(varfunct)`	variance title
`e(varfuncf)`	variance function
`e(link)`	program to calculate link function
`e(linkt)`	link title
`e(linkf)`	link function
`e(m)`	number of binomial trials
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(title_fl)`	family–link title
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(cons)`	`noconstant` or not set
`e(hac_kernel)`	HAC kernel
`e(hac_lag)`	HAC lag
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(opt)`	type of optimization
`e(opt1)`	optimization title, line 1
`e(opt2)`	optimization title, line 2
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(marginsnotok)`	predictions disallowed by `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, *p*-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`binreg, ml` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(df)`	residual degrees of freedom
`e(phi)`	model scale parameter
`e(aic)`	model AIC, if `ml`
`e(bic)`	model BIC
`e(ll)`	log likelihood, if `ml`
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(p)`	*p*-value for model test
`e(deviance)`	deviance
`e(deviance_s)`	scaled deviance
`e(deviance_p)`	Pearson deviance
`e(deviance_ps)`	scaled Pearson deviance
`e(dispers)`	dispersion
`e(dispers_s)`	scaled dispersion
`e(dispers_p)`	Pearson dispersion
`e(dispers_ps)`	scaled Pearson dispersion
`e(vf)`	factor set by `vfactor()`, `1` if not set
`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)`	`binreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(eform)`	`eform()` option implied by `or`, `rr`, `hr`, or `rd`
`e(varfunc)`	program to calculate variance function
`e(varfunct)`	variance title
`e(varfuncf)`	variance function
`e(link)`	program to calculate link function
`e(linkt)`	link title
`e(linkf)`	link function
`e(m)`	number of binomial trials

`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(title_fl)`	family–link title
`e(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(cons)`	`noconstant` or not set
`e(hac_kernel)`	HAC kernel
`e(hac_lag)`	HAC lag
`e(chi2type)`	`Wald`; 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(opt1)`	optimization title, line 1
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`e(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(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

Let π_i be the probability of success for the ith observation, $i = 1, \dots, N$, and let $X\beta$ be the linear predictor. The link function relates the covariates of each observation to its respective probability through the linear predictor.

In logistic regression, the logit link is used:

$$\ln\left(\frac{\pi}{1-\pi}\right) = X\beta$$

The regression coefficient β_k represents the change in the logarithm of the odds associated with a one-unit change in the value of the X_k covariate; thus, $\exp(\beta_k)$ is the ratio of the odds associated with a change of one unit in X_k.

For risk differences, the identity link $\pi = X\beta$ is used. The regression coefficient β_k represents the risk difference associated with a change of one unit in X_k. When using the identity link, you can obtain fitted probabilities outside the interval $(0, 1)$. As suggested by Wacholder, at each iteration, fitted probabilities are checked for range conditions (and put back in range if necessary). For example, if the identity link results in a fitted probability that is smaller than 1e–4, the probability is replaced with 1e–4 before the link function is calculated.

A similar adjustment is made for the logarithmic link, which is used for estimating the risk ratio, $\ln(\pi) = X\beta$, where $\exp(\beta_k)$ is the risk ratio associated with a change of one unit in X_k, and for the log-complement link used to estimate the probability of no disease or health, where $\exp(\beta_k)$ represents the "health ratio" associated with a change of one unit in X_k.

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*.

References

Cummings, P. 2009. Methods for estimating adjusted risk ratios. *Stata Journal* 9: 175–196.

Kleinbaum, D. G., and M. Klein. 2010. *Logistic Regression: A Self-Learning Text.* 3rd ed. New York: Springer. https://doi.org/10.1007/978-1-4419-1742-3.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman and Hall/CRC.

Wacholder, S. 1986. Binomial regression in GLIM: Estimating risk ratios and risk differences. *American Journal of Epidemiology* 123: 174–184. https://doi.org/10.1093/oxfordjournals.aje.a114212.

Wright, J. T., I. G. Barrison, I. G. Lewis, K. D. MacRae, E. J. Waterson, P. J. Toplis, M. G. Gordon, N. F. Morris, and I. M. Murray-Lyon. 1983. Alcohol consumption, pregnancy, and low birthweight. *Lancet* 1: 663–665. https://doi.org/10.1016/S0140-6736(83)91964-5.

Also see

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

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

[BAYES] **bayes: binreg** — Bayesian generalized linear models: Extensions to the binomial family

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

[ME] **meglm** — Multilevel mixed-effects generalized linear models

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

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

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

[U] 20 Estimation and postestimation commands