General use

glm fits generalized linear models of y with covariates $\mathbf{x}$:

$$g\{E(y)\} = \mathbf{x}\boldsymbol{\beta}, \qquad y \sim F$$

In the above, $g(\,)$ is called the link function and F the distributional family. Substituting various definitions for $g(\,)$ and F results in a surprising array of models. For instance, if y is distributed as Gaussian (normal) and $g(\,)$ is the identity function, we have

$$E(y) = \mathbf{x}\boldsymbol{\beta}, \qquad y \sim \text{Normal}$$

or linear regression. If $g(\,)$ is the logit function and y is distributed as Bernoulli, we have

$$\text{logit}\{E(y)\} = \mathbf{x}\boldsymbol{\beta}, \qquad y \sim \text{Bernoulli}$$

or logistic regression. If $g(\,)$ is the natural log function and y is distributed as Poisson, we have

$$\ln\{E(y)\} = \mathbf{x}\boldsymbol{\beta}, \qquad y \sim \text{Poisson}$$

or Poisson regression, also known as the log-linear model. Other combinations are possible.

Although `glm` can be used to perform linear regression (and, in fact, does so by default), this should be viewed as an instructional feature; `regress` produces such estimates more quickly, and numerous post-estimation commands are available to explore the adequacy of the fit; see [R] **regress** and [R] **regression diagnostics**.

In any case, you specify the link function using the `link()` option and the distributional family using `family()`. The allowed link functions are

Link function	glm option
identity	`link(identity)`
log	`link(log)`
logit	`link(logit)`
probit	`link(probit)`
complementary log-log	`link(cloglog)`
odds power	`link(opower #)`
power	`link(power #)`
negative binomial	`link(nbinomial)`
log-log	`link(loglog)`
log-compliment	`link(logc)`

Define $\mu = E(y)$ and $\eta = g(\mu)$, meaning that $g(\cdot)$ maps $E(y)$ to $\eta = \mathbf{x}\boldsymbol{\beta} + \text{offset}$.

Link function `identity` is defined as $\eta = g(\mu) = \mu$.

Link function `log` is defined as $\eta = \ln(\mu)$.

Link function `logit` is defined as $\eta = \ln\{\mu/(1-\mu)\}$, the natural log of the odds.

Link function `probit` is defined as $\eta = \Phi^{-1}(\mu)$, where $\Phi^{-1}(\,)$ is the inverse Gaussian cumulative.

Link function `cloglog` is defined as $\eta = \ln\{-\ln(1-\mu)\}$.

Link function `opower` is defined as $\eta = \left[\{\mu/(1-\mu)\}^n - 1\right]/n$, the power of the odds. The function is generalized so that `link(opower 0)` is equivalent to `link(logit)`, the natural log of the odds.

Link function power is defined as $\eta = \mu^n$. Specifying `link(power 1)` is equivalent to specifying `link(identity)`. The power function is generalized so that $\mu^0 \equiv \ln(\mu)$. Thus, `link(power 0)` is equivalent to `link(log)`. Negative powers are, of course, allowed.

Link function nbinomial is defined as $\eta = \ln\{\mu/(\mu+k)\}$, where $k = 1$ if `family(nbinomial)` is specified and $k = \#_k$ if `family(nbinomial` $\#_k$`)` is specified.

Link function loglog is defined as $\eta = -\ln\{-\ln(\mu)\}$.

Link function logc is defined as $\eta = \ln(1 - \mu)$.

The allowed distributional families are

Family	glm option
Gaussian (normal)	family(gaussian)
inverse Gaussian	family(igaussian)
Bernoulli/binomial	family(binomial)
Poisson	family(poisson)
negative binomial	family(nbinomial)
gamma	family(gamma)

`family(normal)` is allowed as a synonym for `family(gaussian)`.

The binomial distribution can be specified as (1) `family(binomial)`, (2) `family(binomial` $\#_N$`)`, or (3) `family(binomial` *varname*$_N$`)`. In case 2, $\#_N$ is the value of the binomial denominator N, the number of trials. Specifying `family(binomial 1)` is the same as specifying `family(binomial)`; both mean that y has the Bernoulli distribution with values 0 and 1 only. In case 3, *varname*$_N$ is the variable containing the binomial denominator, allowing the number of trials to vary across observations.

The negative binomial distribution can be specified as (1) `family(nbinomial)` or (2) `family(nbinomial` $\#_k$`)`. Omitting $\#_k$ is equivalent to specifying `family(nbinomial 1)`. The value $\#_k$ enters the variance and deviance functions. Typical values range between .01 and 2; see the technical note below.

You do not have to specify both `family()` and `link()`; the default `link()` is the canonical link for the specified `family()` (except for `nbinomial`):

Family	Default link
family(gaussian)	link(identity)
family(igaussian)	link(power -2)
family(binomial)	link(logit)
family(poisson)	link(log)
family(nbinomial)	link(log)
family(gamma)	link(power -1)

If you do specify both `family()` and `link()`, note that not all combinations make sense. You may choose from the following combinations:

	identity	log	logit	probit	cloglog	power	opower	nbinomial	loglog	logc
Gaussian	x	x				x				
inverse Gaussian	x	x				x				
binomial	x	x	x	x	x	x	x		x	x
Poisson	x	x				x				
negative binomial	x	x				x		x		
gamma	x	x				x				

❑ Technical Note

Some family() and link() combinations result in models already fitted by Stata. These are

family()	link()	Options	Other Stata command
gaussian	identity	*nothing* | irls | irls oim	regress
gaussian	identity	t(*var*) nwest(nwest #) vfactor($\#_v$)	newey, t(*var*) lag(#) (see note 1)
binomial	cloglog	*nothing* | irls oim	cloglog (see note 2)
binomial	probit	*nothing* | irls oim	probit (see note 2)
binomial	logit	*nothing* | irls | irls oim	logit or logistic (see note 3)
poisson	log	*nothing* | irls | irls oim	poisson (see note 3)
nbinomial	log	*nothing* | irls oim	nbreg (see note 4)
gamma	log	scale(1)	streg, dist(exp) nohr (see note 5)

Notes:

1. The variance factor $\#_v$ should be set to $n/(n-k)$, where n is the number of observations and k the number of regressors. If not specified, the estimated standard errors will, as a result, differ by this factor.

2. In these cases, since the link is not the canonical link for the binomial family, one must specify the oim option if using irls to get equivalent standard errors. If irls is used without oim, then the regression coefficients will be the same but the standard errors only asymptotically equivalent. If no options are specified (*nothing*), glm will optimize using Newton–Raphson, making it equivalent to the other Stata command.

   See [R] **cloglog** and [R] **probit** for more details about these commands.

3. In these cases, since the canonical link is being used, the standard errors will be equivalent whether the EIM or the OIM estimator of variance is used.

4. Family negative binomial, log-link models—also known as negative binomial regression—are used for data with an overdispersed Poisson distribution. Although glm can be used to fit such models, use of Stata's maximum-likelihood nbreg command is probably better. In the GLM approach, one specifies family(nbinomial $\#_k$) and then searches for a $\#_k$ that results in the deviance-based dispersion being 1. nbreg, on the other hand, finds the maximum likelihood estimate of $\#_k$ and reports a confidence interval for it; see [R] **nbreg** and Rogers (1993). Of course, glm allows links other than log, and for those links, including the canonical nbinomial link, you will need to use glm. Since the default link for family(nbinomial) is a noncanonical link, standard errors will be only asymptotically equivalent if glm, irls without the oim option is used.

5. glm can be used to estimate parameters from exponential regressions, but this requires specifying scale(1). However, censoring is not available with this method. Censored exponential regression may be modeled using glm with family(poisson). The log of the original response is entered into a Poisson model as an offset, while the new response is the censor variable. The result of such modeling is identical to the log relative hazard parameterization of streg, dist(exp) nohr. See [ST] **streg** for details about the streg command.

In general, where there is overlap between a capability of glm and that of some other Stata command, we recommend using the other Stata command. Our recommendation is not due to some inferiority of the GLM approach. Rather, it is that those other, more specialized commands, by being specialized, provide options and ancillary commands missing in the broader glm framework. Nevertheless, glm does produce the same answers where it should.

Special note. In cases where equivalence is expected, for some datasets, one may still see very slight differences in the results, most often only in the latter digits of the standard errors. When comparing `glm` output to an "equivalent" Stata command, these tiny discrepancies arise for many reasons:

a. `glm` uses a general methodology for starting values, while the equivalent Stata command may be more specialized in its treatment of starting values.

b. When using a canonical link, `glm, irls` should be equivalent to the maximum likelihood method of the equivalent Stata command, yet the convergence criterion is different (one is in terms of deviance, the other in terms of log likelihood). These discrepancies are easily resolved by adjusting one convergence criterion to correspond to the other.

c. In cases where both `glm` and the equivalent Stata command are using Newton–Raphson, small differences may still occur if the Stata command has a different default convergence criterion than `glm`. Again, adjusting the convergence criterion will resolve the difference. See [R] **ml** and [R] **maximize** for more details. ❑

▷ Example

In [R] **logistic**, we fit a model based on data from a study of risk factors associated with low birth weight (Hosmer and Lemeshow 2000, 25). We can replicate the estimation using `glm`:

```
. use http://www.stata-press.com/data/r8/lbw
(Hosmer & Lemeshow data)

. xi: glm low age lwt i.race smoke ptl ht ui, f(bin) l(logit)
i.race            _Irace_1-3          (naturally coded; _Irace_1 omitted)

Iteration 0:   log likelihood =  -101.0213
Iteration 1:   log likelihood = -100.72519
Iteration 2:   log likelihood =   -100.724
Iteration 3:   log likelihood =   -100.724

Generalized linear models                          No. of obs      =       189
Optimization     : ML: Newton-Raphson              Residual df     =       180
                                                   Scale parameter =         1
Deviance         =  201.4479911                    (1/df) Deviance =  1.119156
Pearson          =  182.0233425                    (1/df) Pearson  =  1.011241

Variance function: V(u) = u*(1-u)                  [Bernoulli]
Link function    : g(u) = ln(u/(1-u))              [Logit]
Standard errors  : OIM

Log likelihood   = -100.7239956                    AIC             =    1.1611
BIC              = -742.0664716

------------------------------------------------------------------------------
         low |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |  -.0271003   .0364504    -0.74   0.457    -.0985418    .0443412
         lwt |  -.0151508   .0069259    -2.19   0.029    -.0287253   -.0015763
    _Irace_2 |   1.262647   .5264101     2.40   0.016     .2309024    2.294392
    _Irace_3 |   .8620792   .4391532     1.96   0.050     .0013548    1.722804
       smoke |   .9233448   .4008266     2.30   0.021      .137739    1.708951
         ptl |   .5418366    .346249     1.56   0.118     -.136799    1.220472
          ht |   1.832518   .6916292     2.65   0.008     .4769494    3.188086
          ui |   .7585135   .4593768     1.65   0.099    -.1418484    1.658875
       _cons |   .4612239    1.20459     0.38   0.702    -1.899729    2.822176
------------------------------------------------------------------------------
```

`glm`, by default, presents coefficient estimates, whereas `logistic` presents the exponentiated coefficients—the odds ratios. `glm`'s `eform` option reports exponentiated coefficients, and `glm`, like Stata's other estimation commands, replays results.

```
. glm, eform

Generalized linear models                          No. of obs      =       189
Optimization     : ML: Newton-Raphson              Residual df     =       180
                                                   Scale parameter =         1
Deviance         =  201.4479911                    (1/df) Deviance =  1.119156
Pearson          =  182.0233425                    (1/df) Pearson  =  1.011241

Variance function: V(u) = u*(1-u)                  [Bernoulli]
Link function    : g(u) = ln(u/(1-u))              [Logit]
Standard errors  : OIM

Log likelihood   = -100.7239956                    AIC             =    1.1611
BIC              = -742.0664716

------------------------------------------------------------------------------
         low | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .9732636   .0354759    -0.74   0.457     .9061578    1.045339
         lwt |   .9849634   .0068217    -2.19   0.029     .9716834    .9984249
    _Irace_2 |   3.534767   1.860737     2.40   0.016     1.259736    9.918406
    _Irace_3 |   2.368079   1.039949     1.96   0.050     1.001356    5.600207
       smoke |   2.517698    1.00916     2.30   0.021     1.147676    5.523162
         ptl |   1.719161   .5952579     1.56   0.118     .8721455    3.388787
          ht |   6.249602   4.322408     2.65   0.008     1.611152    24.24199
          ui |     2.1351   .9808153     1.65   0.099     .8677528      5.2534
------------------------------------------------------------------------------
```

These results are the same as reported in [R] **logistic**.

Included in the output header are values for the Akaike (1973) information criterion (AIC) and the Bayesian information criterion (BIC) (Raftery 1996). Both are measures of model fit adjusted for the number of parameters that can be compared across models. In both cases, a smaller value generally indicates a better model fit. AIC is based on the log likelihood, and thus is only available when Newton–Raphson optimization is employed. BIC is based on the deviance, and thus is always available.

◁