nbreg — Negative binomial regression

Description

`nbreg` fits a negative binomial regression model for a nonnegative count dependent variable. In this model, the count variable is believed to be generated by a Poisson-like process, except that the variation is allowed to be greater than that of a true Poisson. This extra variation is referred to as overdispersion.

`gnbreg` fits a generalization of the negative binomial mean-dispersion model; the shape parameter α may also be parameterized.

Quick start

Negative binomial model of `y` on `x1` and categorical variable `a`

```
nbreg y x1 i.a
```

Same as above, but report results as incidence-rate ratios

```
nbreg y x1 i.a, irr
```

Same as above, and specify exposure variable `evar`

```
nbreg y x1 i.a, irr exposure(evar)
```

Generalized negative binomial model with shape parameter α a function of `x2` and `x3`

```
gnbreg y x1 i.a, lnalpha(x2 x3)
```

Add log of exposure, `lnevar`, as an offset

```
gnbreg y x1 i.a, lnalpha(x2 x3) offset(lnevar)
```

Menu

nbreg

Statistics > Count outcomes > Negative binomial regression

gnbreg

Statistics > Count outcomes > Generalized negative binomial regression

Syntax

Negative binomial regression model

nbreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *nbreg_options*]

Generalized negative binomial model

gnbreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *gnbreg_options*]

nbreg_options	Description
Model	
noconstant	suppress constant term
dispersion(mean)	parameterization of dispersion; the default
dispersion(constant)	constant dispersion for all observations
exposure(*varname*$_e$)	include ln(*varname*$_e$) in model with coefficient constrained to 1
offset(*varname*$_o$)	include *varname*$_o$ in model with coefficient constrained to 1
constraints(*constraints*)	apply specified linear constraints
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, opg, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
nolrtest	suppress likelihood-ratio test
irr	report incidence-rate ratios
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	
maximize_options	control the maximization process; seldom used
collinear	keep collinear variables
coeflegend	display legend instead of statistics

gnbreg_options	Description
Model	
noconstant	suppress constant term
lnalpha(*varlist*)	dispersion model variables
exposure(*varname*$_e$)	include ln(*varname*$_e$) in model with coefficient constrained to 1
offset(*varname*$_o$)	include *varname*$_o$ in model with coefficient constrained to 1
constraints(*constraints*)	apply specified linear constraints
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, opg, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
irr	report incidence-rate ratios
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	
maximize_options	control the maximization process; seldom used
collinear	keep collinear variables
coeflegend	display legend instead of statistics

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

depvar, *indepvars*, *varname*$_e$, and *varname*$_o$ may contain time-series operators (nbreg only); see [U] **11.4.4 Time-series varlists**.

bayes, bayesboot (nbreg only), bootstrap, by (nbreg only), collect, fmm (nbreg only), fp (nbreg only), jackknife, mfp (nbreg only), mi estimate, nestreg (nbreg only), rolling, statsby, stepwise, and svy are allowed; see [U] **11.1.10 Prefix commands**. For more details, see [BAYES] **bayes: gnbreg**, [BAYES] **bayes: nbreg**, and [FMM] **fmm: nbreg**.

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

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

vce() and weights are not allowed with the svy prefix; see [SVY] **svy**.

fweights, iweights, and pweights 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 for nbreg

Model

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

dispersion(mean | constant) specifies the parameterization of the model. dispersion(mean), the default, yields a model with dispersion equal to $1 + \alpha \exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j)$; that is, the dispersion is a function of the expected mean: $\exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j)$. dispersion(constant) has dispersion equal to $1 + \delta$; that is, it is a constant for all observations.

exposure(*varname*$_e$), offset(*varname*$_o$), constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

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

Reporting

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

nolrtest suppresses fitting the Poisson model. Without this option, a comparison Poisson model is fit, and the likelihood is used in a likelihood-ratio test of the null hypothesis that the dispersion parameter is zero.

irr reports estimated coefficients transformed to incidence-rate ratios, that is, e^{β_i} rather than β_i. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated or stored. irr may be specified at estimation or when replaying previously estimated results.

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

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

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

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

collinear, coeflegend; see [R] **Estimation options**.

Options for gnbreg

Model

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

lnalpha(*varlist*) allows you to specify a linear equation for $\ln\alpha$. Specifying lnalpha(male old) means that $\ln\alpha = \gamma_0 + \gamma_1$male $+ \gamma_2$old, where γ_0, γ_1, and γ_2 are parameters to be estimated along with the other model coefficients. If this option is not specified, gnbreg and nbreg will produce the same results because the shape parameter will be parameterized as a constant.

exposure(*varname$_e$*), offset(*varname$_o$*), constraints(*constraints*); see [R] **Estimation options**.

SE/Robust

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

Reporting

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

irr reports estimated coefficients transformed to incidence-rate ratios, that is, e^{β_i} rather than β_i. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated or stored. irr may be specified at estimation or when replaying previously estimated results.

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

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

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

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

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction to negative binomial regression
nbreg
gnbreg

Introduction to negative binomial regression

Negative binomial regression models the number of occurrences (counts) of an event when the event has extra-Poisson variation, that is, when it has overdispersion. The Poisson regression model is

$$y_j \sim \text{Poisson}(\mu_j)$$

where

$$\mu_j = \exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j)$$

for observed counts y_j with covariates $\mathbf{x}_j$ for the jth observation. One derivation of the negative binomial mean-dispersion model is that individual units follow a Poisson regression model, but there is an omitted variable ζ_j, such that e^{ζ_j} follows a gamma distribution with mean 1 and variance α:

$$y_j \sim \text{Poisson}(\mu_j^*)$$

where

$$\mu_j^* = \exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j + \zeta_j)$$

and

$$e^{\zeta_j} \sim \text{Gamma}(1/\alpha, \alpha)$$

With this parameterization, a Gamma(a, b) distribution will have expectation ab and variance ab^2.

We refer to α as the overdispersion parameter. The larger α is, the greater the overdispersion. The Poisson model corresponds to $\alpha = 0$. `nbreg` parameterizes α as $\ln\alpha$. `gnbreg` allows $\ln\alpha$ to be modeled as $\ln\alpha_j = \mathbf{z}_j\boldsymbol{\gamma}$, a linear combination of covariates $\mathbf{z}_j$.

`nbreg` will fit two different parameterizations of the negative binomial model. The default, described above and also given by the `dispersion(mean)` option, has dispersion for the jth observation equal to $1 + \alpha\exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j)$. This is seen by noting that the above implies that

$$\mu_j^* \sim \text{Gamma}(1/\alpha, \alpha\mu_j)$$

and thus

$$\begin{aligned}\text{Var}(y_j) &= E\left\{\text{Var}(y_j|\mu_j^*)\right\} + \text{Var}\left\{E(y_j|\mu_j^*)\right\} \\ &= E(\mu_j^*) + \text{Var}(\mu_j^*) \\ &= \mu_j(1 + \alpha\mu_j)\end{aligned}$$

The alternative parameterization, given by the `dispersion(constant)` option, has dispersion equal to $1 + \delta$; that is, it is constant for all observations. This is so because the constant-dispersion model assumes instead that

$$\mu_j^* \sim \text{Gamma}(\mu_j/\delta, \delta)$$

and thus $\text{Var}(y_j) = \mu_j(1 + \delta)$. The Poisson model corresponds to $\delta = 0$.

For detailed derivations of both models, see Cameron and Trivedi (2013, 80–89). In particular, note that the mean-dispersion model is known as the NB2 model in their terminology, whereas the constant-dispersion model is referred to as the NB1 model.

See Long and Freese (2014) and Cameron and Trivedi (2022, chap. 20) for a discussion of the negative binomial regression model with Stata examples and for a discussion of other regression models for count data.

Hilbe (2011) provides an extensive review of the negative binomial model and its variations, using Stata examples.

nbreg

It is not uncommon to posit a Poisson regression model and observe a lack of model fit. The following data appeared in Rodríguez (1993):

```
. use https://www.stata-press.com/data/r19/rod93
. list, sepby(cohort)
```

	cohort	age_mos	deaths	exposure
1.	1941–1949	0.5	168	278.4
2.	1941–1949	2.0	48	538.8
3.	1941–1949	4.5	63	794.4
4.	1941–1949	9.0	89	1,550.8
5.	1941–1949	18.0	102	3,006.0
6.	1941–1949	42.0	81	8,743.5
7.	1941–1949	90.0	40	14,270.0
8.	1960–1967	0.5	197	403.2
9.	1960–1967	2.0	48	786.0
10.	1960–1967	4.5	62	1,165.3
11.	1960–1967	9.0	81	2,294.8
12.	1960–1967	18.0	97	4,500.5
13.	1960–1967	42.0	103	13,201.5
14.	1960–1967	90.0	39	19,525.0
15.	1968–1976	0.5	195	495.3
16.	1968–1976	2.0	55	956.7
17.	1968–1976	4.5	58	1,381.4
18.	1968–1976	9.0	85	2,604.5
19.	1968–1976	18.0	87	4,618.5
20.	1968–1976	42.0	70	9,814.5
21.	1968–1976	90.0	10	5,802.5

```
. generate logexp = ln(exposure)
```

```
. poisson deaths i.cohort, offset(logexp)
Iteration 0:  Log likelihood = -2160.0544
Iteration 1:  Log likelihood = -2159.5162
Iteration 2:  Log likelihood = -2159.5159
Iteration 3:  Log likelihood = -2159.5159
Poisson regression                                      Number of obs =     21
                                                        LR chi2(2)    =  49.16
                                                        Prob > chi2   = 0.0000
Log likelihood = -2159.5159                             Pseudo R2     = 0.0113

------------------------------------------------------------------------------
      deaths | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      cohort |
   1960-1967 |  -.3020405   .0573319    -5.27   0.000    -.4144089   -.1896721
   1968-1976 |   .0742143   .0589726     1.26   0.208    -.0413698    .1897983
             |
       _cons |  -3.899488   .0411345   -94.80   0.000     -3.98011   -3.818866
      logexp |          1  (offset)
------------------------------------------------------------------------------

. estat gof
         Deviance goodness-of-fit =  4190.689
         Prob > chi2(18)          =    0.0000
         Pearson goodness-of-fit  =  15387.67
         Prob > chi2(18)          =    0.0000
```

The extreme significance of the goodness-of-fit χ^2 indicates that the Poisson regression model is inappropriate, suggesting to us that we should try a negative binomial model:

```
. nbreg deaths i.cohort, offset(logexp) nolog
Negative binomial regression                            Number of obs =     21
                                                        LR chi2(2)    =   0.40
Dispersion: mean                                        Prob > chi2   = 0.8171
Log likelihood = -131.3799                              Pseudo R2     = 0.0015

------------------------------------------------------------------------------
      deaths | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      cohort |
   1960-1967 |  -.2676187   .7237203    -0.37   0.712    -1.686084    1.150847
   1968-1976 |  -.4573957   .7236651    -0.63   0.527    -1.875753    .9609618
             |
       _cons |  -2.086731    .511856    -4.08   0.000     -3.08995   -1.083511
      logexp |          1  (offset)
-------------+----------------------------------------------------------------
    /lnalpha |   .5939963   .2583615                      .0876171    1.100376
-------------+----------------------------------------------------------------
       alpha |   1.811212   .4679475                       1.09157    3.005295
------------------------------------------------------------------------------
LR test of alpha=0: chibar2(01) = 4056.27              Prob >= chibar2 = 0.000
```

Our original Poisson model is a special case of the negative binomial—it corresponds to $\alpha = 0$. `nbreg`, however, estimates α indirectly, estimating instead $\ln\alpha$. In our model, $\ln\alpha = 0.594$, meaning that $\alpha = 1.81$ (`nbreg` undoes the transformation for us at the bottom of the output).

To test $\alpha = 0$ (equivalent to $\ln\alpha = -\infty$), `nbreg` performs a likelihood-ratio test. The staggering χ^2 value of 4,056 asserts that the probability that we would observe these data conditional on $\alpha = 0$ is virtually zero, that is, conditional on the process being Poisson. The data are not Poisson. It is not accidental that this χ^2 value is close to the goodness-of-fit statistic from the Poisson regression itself.

❑ Technical note

The usual Gaussian test of $\alpha = 0$ is omitted because this test occurs on the boundary, invalidating the usual theory associated with such tests. However, the likelihood-ratio test of $\alpha = 0$ has been modified to be valid on the boundary. In particular, the null distribution of the likelihood-ratio test statistic is not the usual χ_1^2, but rather a 50 : 50 mixture of a χ_0^2 (point mass at zero) and a χ_1^2, denoted as $\overline{\chi}_{01}^2$. See Gutierrez, Carter, and Drukker (2001) for more details.

❑

❑ Technical note

The negative binomial model deals with cases in which there is more variation than would be expected if the process were Poisson. The negative binomial model is not helpful if there is less than Poisson variation—if the variance of the count variable is less than its mean. However, underdispersion is uncommon. Poisson models arise because of independently generated events. Overdispersion comes about if some of the parameters (causes) of the Poisson processes are unknown. To obtain underdispersion, the sequence of events somehow would have to be regulated; that is, events would not be independent but controlled based on past occurrences.

❑

gnbreg

`gnbreg` is a generalization of `nbreg`, `dispersion(mean)`. Whereas in `nbreg`, one $\ln\alpha$ is estimated, `gnbreg` allows $\ln\alpha$ to vary, observation by observation, as a linear combination of another set of covariates: $\ln\alpha_j = \mathbf{z}_j\gamma$.

We will assume that the number of deaths is a function of age, whereas the $\ln\alpha$ parameter is a function of cohort. To fit the model, we type

```
. gnbreg deaths age_mos, lnalpha(i.cohort) offset(logexp)
Fitting constant-only model:
Iteration 0:  Log likelihood =   -187.067  (not concave)
Iteration 1:  Log likelihood = -138.13047
Iteration 2:  Log likelihood = -133.83164
Iteration 3:  Log likelihood = -131.59551
Iteration 4:  Log likelihood =  -131.5795
Iteration 5:  Log likelihood = -131.57948
Iteration 6:  Log likelihood = -131.57948
Fitting full model:
Iteration 0:  Log likelihood = -124.34327
Iteration 1:  Log likelihood = -117.76701
Iteration 2:  Log likelihood = -117.56403
Iteration 3:  Log likelihood = -117.56164
Iteration 4:  Log likelihood = -117.56164
Generalized negative binomial regression                Number of obs =     21
                                                        LR chi2(1)    =  28.04
                                                        Prob > chi2   = 0.0000
Log likelihood = -117.56164                             Pseudo R2     = 0.1065
```

deaths	Coefficient	Std. err.	z	P>|z|	[95% conf. interval]	
deaths						
age_mos	-.0516657	.0051747	-9.98	0.000	-.061808	-.0415233
_cons	-1.867225	.2227944	-8.38	0.000	-2.303894	-1.430556
logexp	1	(offset)				
lnalpha						
cohort						
1960–1967	.0939546	.7187747	0.13	0.896	-1.314818	1.502727
1968–1976	.0815279	.7365476	0.11	0.912	-1.362079	1.525135
_cons	-.4759581	.5156502	-0.92	0.356	-1.486614	.5346978

We find that age is a significant determinant of the number of deaths. The standard errors for the variables in the $\ln\alpha$ equation suggest that the overdispersion parameter does not vary across cohorts. We can test this assertion by typing

```
. test 2.cohort 3.cohort
 ( 1)  [lnalpha]2.cohort = 0
 ( 2)  [lnalpha]3.cohort = 0
           chi2(  2) =    0.02
         Prob > chi2 =    0.9904
```

There is no evidence of variation by cohort in these data.

❏ Technical note

Note the intentional absence of a likelihood-ratio test for $\alpha = 0$ in gnbreg. The test is affected by the same boundary condition that affects the comparison test in nbreg; however, when α is parameterized by more than a constant term, the null distribution becomes intractable. For this reason, we recommend using nbreg to test for overdispersion and, if you have reason to believe that overdispersion exists, only then modeling the overdispersion using gnbreg.

❏

Stored results

nbreg stores the following in e():

Scalars	
e(N)	number of observations
e(k)	number of parameters
e(k_aux)	number of auxiliary 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^2
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(ll_c)	log likelihood, comparison model
e(alpha)	value of alpha
e(delta)	value of delta
e(N_clust)	number of clusters
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(p)	p-value for model test
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise
Macros	
e(cmd)	nbreg
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(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(dispers)	mean or constant
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict

e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices

e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

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.

gnbreg 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(r2_p)	pseudo-R^2
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(N_clust)	number of clusters
e(chi2)	χ^2
e(p)	p-value for model test
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	gnbreg
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(offset1)	linear offset variable
e(chi2type)	Wald or LR; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V

`e(predict)`	program used to implement `predict`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log (up to 20 iterations)
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
Functions	
`e(sample)`	marks estimation sample

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

See [R] **poisson** and Johnson, Kemp, and Kotz (2005, chap. 4) for an introduction to the Poisson distribution.

Methods and formulas are presented under the following headings:

Mean-dispersion model
Constant-dispersion model

Mean-dispersion model

A negative binomial distribution can be regarded as a gamma mixture of Poisson random variables. The number of times something occurs, y_j, is distributed as Poisson$(\nu_j\mu_j)$. That is, its conditional likelihood is

$$f(y_j \mid \nu_j) = \frac{(\nu_j\mu_j)^{y_j} e^{-\nu_j\mu_j}}{\Gamma(y_j+1)}$$

where $\mu_j = \exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j)$ and ν_j is an unobserved parameter with a Gamma$(1/\alpha, \alpha)$ density:

$$g(\nu) = \frac{\nu^{(1-\alpha)/\alpha} e^{-\nu/\alpha}}{\alpha^{1/\alpha}\Gamma(1/\alpha)}$$

This gamma distribution has mean 1 and variance α, where α is our ancillary parameter.

The unconditional likelihood for the jth observation is therefore

$$f(y_j) = \int_0^\infty f(y_j \mid \nu) g(\nu)\, d\nu = \frac{\Gamma(m+y_j)}{\Gamma(y_j+1)\Gamma(m)}\, p_j^m (1-p_j)^{y_j}$$

where $p_j = 1/(1+\alpha\mu_j)$ and $m = 1/\alpha$. Solutions for α are handled by searching for $\ln\alpha$ because α must be greater than zero.

The log likelihood (with weights w_j and offsets) is given by

$$m = 1/\alpha \qquad p_j = 1/(1+\alpha\mu_j) \qquad \mu_j = \exp(\mathbf{x}_j\boldsymbol{\beta} + \text{offset}_j)$$

$$\begin{aligned}\ln L = \sum_{j=1}^{n} w_j \Big[& \ln\{\Gamma(m+y_j)\} - \ln\{\Gamma(y_j+1)\} \\ & - \ln\{\Gamma(m)\} + m\ln(p_j) + y_j \ln(1-p_j)\Big]\end{aligned}$$

For gnbreg, α can vary across the observations according to the parameterization $\ln\alpha_j = \mathbf{z}_j\boldsymbol{\gamma}$.

Constant-dispersion model

The constant-dispersion model assumes that y_j is conditionally distributed as Poisson(μ_j^*), where $\mu_j^* \sim \text{Gamma}(\mu_j/\delta, \delta)$ for some dispersion parameter δ (by contrast, the mean-dispersion model assumes that $\mu_j^* \sim \text{Gamma}(1/\alpha, \alpha\mu_j)$). The log likelihood is given by

$$m_j = \mu_j/\delta \qquad p = 1/(1+\delta)$$

$$\begin{aligned}\ln L = \sum_{j=1}^{n} w_j \Big[& \ln\{\Gamma(m_j+y_j)\} - \ln\{\Gamma(y_j+1)\} \\ & - \ln\{\Gamma(m_j)\} + m_j\ln(p) + y_j \ln(1-p)\Big]\end{aligned}$$

with everything else defined as before in the calculations for the mean-dispersion model.

`nbreg` and `gnbreg` support 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*.

These commands also support estimation with survey data. For details on VCEs with survey data, see [SVY] **Variance estimation**.

References

Cameron, A. C., and P. K. Trivedi. 2013. *Regression Analysis of Count Data*. 2nd ed. New York: Cambridge University Press.

———. 2022. *Microeconometrics Using Stata*. 2nd ed. College Station, TX: Stata Press.

Cummings, T. H., J. W. Hardin, A. C. McLain, J. R. Hussey, K. J. Bennett, and G. M. Wingood. 2015. Modeling heaped count data. *Stata Journal* 15: 457–479.

Deb, P., E. C. Norton, and W. G. Manning. 2017. *Health Econometrics Using Stata*. College Station, TX: Stata Press.

Deb, P., and P. K. Trivedi. 2006. Maximum simulated likelihood estimation of a negative binomial regression model with multinomial endogenous treatment. *Stata Journal* 6: 246–255.

Gutierrez, R. G., S. L. Carter, and D. M. Drukker. 2001. sg160: On boundary-value likelihood-ratio tests. *Stata Technical Bulletin* 60: 15–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 269–273. College Station, TX: Stata Press.

Harris, T., J. M. Hilbe, and J. W. Hardin. 2014. Modeling count data with generalized distributions. *Stata Journal* 14: 562–579.

Hilbe, J. M. 2011. *Negative Binomial Regression*. 2nd ed. Cambridge: Cambridge University Press.

———. 2014. *Modeling Count Data*. New York: Cambridge University Press.

Johnson, N. L., A. W. Kemp, and S. Kotz. 2005. *Univariate Discrete Distributions*. 3rd 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. 2001. Predicted probabilities for count models. *Stata Journal* 1: 51–57.

———. 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.

Rodríguez, G. 1993. sbe10: An improvement to poisson. *Stata Technical Bulletin* 11: 11–14. Reprinted in *Stata Technical Bulletin Reprints*, vol. 2, pp. 94–98. College Station, TX: Stata Press.

Xu, X., and J. W. Hardin. 2016. Regression models for bivariate count outcomes. *Stata Journal* 16: 301–315.

Also see

[R] **nbreg postestimation** — Postestimation tools for nbreg and gnbreg

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

[R] **npregress kernel** — Nonparametric kernel regression

[R] **npregress series** — Nonparametric series regression

[R] **poisson** — Poisson regression

[R] **tnbreg** — Truncated negative binomial regression

[R] **zinb** — Zero-inflated negative binomial regression

[BAYES] **bayes: gnbreg** — Bayesian generalized negative binomial regression

[BAYES] **bayes: nbreg** — Bayesian negative binomial regression

[FMM] **fmm: nbreg** — Finite mixtures of negative binomial regression models

[ME] **menbreg** — Multilevel mixed-effects negative binomial regression

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

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

[XT] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

[U] 20 Estimation and postestimation commands