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 $\alpha$ may also be parameterized.

Quick start

Negative binomial model of $y$ on $x_1$ and categorical variable $a$

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

As above, but report results as incidence-rate ratios

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

As above, and specify exposure variable $evar$

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

Generalized negative binomial model with shape parameter $\alpha$ a function of $x_2$ and $x_3$

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

Add log of exposure, $\ln evar$, 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

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noconstant</td>
<td>suppress constant term</td>
</tr>
<tr>
<td>dispersion(mean)</td>
<td>parameterization of dispersion; the default</td>
</tr>
<tr>
<td>dispersion(constant)</td>
<td>constant dispersion for all observations</td>
</tr>
<tr>
<td>exposure(varname)</td>
<td>include ( \ln(varname) ) in model with coefficient constrained to 1</td>
</tr>
<tr>
<td>offset(varname)</td>
<td>include ( varname ) in model with coefficient constrained to 1</td>
</tr>
<tr>
<td>constraints(constraints)</td>
<td>apply specified linear constraints</td>
</tr>
</tbody>
</table>

#### SE/Robust

```
vce(vcetype)
```

\( vcetype \) may be \( oim, robust, cluster clustvar, opg, bootstrap, \) or \( jackknife \)

#### Reporting

```
level(#)
nolrtest
irr
nocnsreport
display_options
```

set confidence level; default is \( level(95) \)
suppress likelihood-ratio test
report incidence-rate ratios
do not display constraints
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
### nbreg — Negative binomial regression

#### gn breg_options

<table>
<thead>
<tr>
<th>Model</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>noconstant</td>
<td>suppress constant term</td>
</tr>
<tr>
<td>lnalpha(varlist)</td>
<td>dispersion model variables</td>
</tr>
<tr>
<td>exposure(varname_e)</td>
<td>include ln(varname_e) in model with coefficient constrained to 1</td>
</tr>
<tr>
<td>offset(varname_o)</td>
<td>include varname_o in model with coefficient constrained to 1</td>
</tr>
<tr>
<td>constraints(constraints)</td>
<td>apply specified linear constraints</td>
</tr>
</tbody>
</table>

#### SE/Robust

| vce(vcetype)              | vcetype may be oim, robust, cluster clustvar, opg, bootstrap, or jackknife  |

#### Reporting

<table>
<thead>
<tr>
<th>level(#)</th>
<th>set confidence level; default is level(95)</th>
</tr>
</thead>
<tbody>
<tr>
<td>irr</td>
<td>report incidence-rate ratios</td>
</tr>
<tr>
<td>nocnsrreport</td>
<td>do not display constraints</td>
</tr>
<tr>
<td>display_options</td>
<td>control columns and column formats, row spacing, line width,</td>
</tr>
<tr>
<td></td>
<td>display of omitted variables and base and empty cells, and</td>
</tr>
<tr>
<td></td>
<td>factor-variable labeling</td>
</tr>
</tbody>
</table>

#### Maximization

<table>
<thead>
<tr>
<th>maximize_options</th>
<th>control the maximization process; seldom used</th>
</tr>
</thead>
<tbody>
<tr>
<td>collinear</td>
<td>keep collinear variables</td>
</tr>
<tr>
<td>coeflegend</td>
<td>display legend instead of statistics</td>
</tr>
</tbody>
</table>

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, bootstrap, by (nbreg only), 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 $1 + \alpha \exp(x_j\beta + \text{offset}_j)$; that is, the dispersion is a function of the expected mean: $\exp(x_j\beta + \text{offset}_j)$. 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.
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.

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, $e^{\beta_i}$ rather than $\beta_i$. 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.

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

noconstant; see [R] Estimation options.

lnalpha(varlist) allows you to specify a linear equation for $\ln \alpha$. Specifying lnalpha(male old) means that $\ln \alpha = \gamma_0 + \gamma_1 \text{male} + \gamma_2 \text{old}$, where $\gamma_0$, $\gamma_1$, and $\gamma_2$ 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.
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, \(e^{\beta_i}\) rather than \(\beta_i\). 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.

cnsreport; see [R] Estimation options.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, noflabel, 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(#), rtolerance(#), 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 stata.com

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

\[ \mu_j = \exp(x_j \beta + \text{offset}_j) \]
for observed counts $y_j$ with covariates $x_j$ for the $j$th 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 $\nu_j$, such that $e^{\nu_j}$ follows a gamma distribution with mean 1 and variance $\alpha$:

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

where

$$\mu_j^* = \exp(x_j \beta + \text{offset}_j + \nu_j)$$

and

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

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

We refer to $\alpha$ as the Gamma($a, b$) distribution will have expectation $ab$ and variance $ab^2$.

The Poisson model corresponds to $\alpha = 0$. `nbreg` parameterizes $\alpha$ as $\ln \alpha$. `gnbreg` allows $\ln \alpha$ to be modeled as $\ln \alpha_j = z_j \gamma$, a linear combination of covariates $z_j$.

`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 $j$th observation equal to $1 + \alpha \exp(x_j \beta + \text{offset}_j)$. This is seen by noting that the above implies that

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

and thus

$$\text{Var}(y_j) = E\{ \text{Var}(y_j | \mu_j^*) \} + \text{Var}\{ E(y_j | \mu_j^*) \} = E(\mu_j^*) + \text{Var}(\mu_j^*) = \mu_j(1 + \alpha \mu_j)$$

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 (2010, chap. 17) 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.
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/r16/rod93
list, sepby(cohort)
```

<table>
<thead>
<tr>
<th>cohort</th>
<th>age_mos</th>
<th>deaths</th>
<th>exposure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1941-1949</td>
<td>0.5</td>
<td>168</td>
<td>278.4</td>
</tr>
<tr>
<td>1941-1949</td>
<td>2.0</td>
<td>48</td>
<td>538.8</td>
</tr>
<tr>
<td>1941-1949</td>
<td>4.5</td>
<td>63</td>
<td>794.4</td>
</tr>
<tr>
<td>1941-1949</td>
<td>9.0</td>
<td>89</td>
<td>1,550.8</td>
</tr>
<tr>
<td>1941-1949</td>
<td>18.0</td>
<td>102</td>
<td>3,006.0</td>
</tr>
<tr>
<td>1941-1949</td>
<td>42.0</td>
<td>81</td>
<td>8,743.5</td>
</tr>
<tr>
<td>1941-1949</td>
<td>90.0</td>
<td>40</td>
<td>14,270.0</td>
</tr>
<tr>
<td>1960-1967</td>
<td>0.5</td>
<td>197</td>
<td>403.2</td>
</tr>
<tr>
<td>1960-1967</td>
<td>2.0</td>
<td>48</td>
<td>786.0</td>
</tr>
<tr>
<td>1960-1967</td>
<td>4.5</td>
<td>62</td>
<td>1,165.3</td>
</tr>
<tr>
<td>1960-1967</td>
<td>9.0</td>
<td>81</td>
<td>2,294.8</td>
</tr>
<tr>
<td>1960-1967</td>
<td>18.0</td>
<td>97</td>
<td>4,500.5</td>
</tr>
<tr>
<td>1960-1967</td>
<td>42.0</td>
<td>103</td>
<td>13,201.5</td>
</tr>
<tr>
<td>1960-1967</td>
<td>90.0</td>
<td>39</td>
<td>19,525.0</td>
</tr>
<tr>
<td>1968-1976</td>
<td>0.5</td>
<td>195</td>
<td>495.3</td>
</tr>
<tr>
<td>1968-1976</td>
<td>2.0</td>
<td>55</td>
<td>956.7</td>
</tr>
<tr>
<td>1968-1976</td>
<td>4.5</td>
<td>58</td>
<td>1,381.4</td>
</tr>
<tr>
<td>1968-1976</td>
<td>9.0</td>
<td>85</td>
<td>2,604.5</td>
</tr>
<tr>
<td>1968-1976</td>
<td>18.0</td>
<td>87</td>
<td>4,618.5</td>
</tr>
<tr>
<td>1968-1976</td>
<td>42.0</td>
<td>70</td>
<td>9,814.5</td>
</tr>
<tr>
<td>1968-1976</td>
<td>90.0</td>
<td>10</td>
<td>5,802.5</td>
</tr>
</tbody>
</table>

```
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 | Coef. | 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) |           |     |     |                     |
The extreme significance of the goodness-of-fit $\chi^2$ indicates that the Poisson regression model is inappropriate, suggesting to us that we should try a negative binomial model:

\[
.\ nbreg \ deaths \ i.\cohorts, \ offset(\text{logexp}) \ nolog
\]

| deaths | Coef.  | 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  | -.2086731 | .511856  | -4.08 | 0.000 | -3.08995 -1.083511   |
| logexp |        |           |      |       |                      |
| 1 (offset) | .5939963 | .2583615 | .0876171 | 1.100376 |

Our original Poisson model is a special case of the negative binomial—it corresponds to $\alpha = 0$. \nbreg, however, estimates $\alpha$ 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 $\chi^2$ 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 $\chi^2$ value is close to the goodness-of-fit statistic from the Poisson regression itself.

\[ LR \ test \ of \ alpha=0: \ chibar2(01) = 4056.27 \quad \text{Prob} >= \ chibar2 = 0.000 \]

\section*{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 $\chi^2$, but rather a 50:50 mixture of a $\chi^2_0$ (point mass at zero) and a $\chi^2_1$, denoted as $\chi^2_{01}$. See Gutierrez, Carter, and Drukker (2001) for more details.

\section*{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 lnα is estimated, gnbreg allows lnα to vary, observation by observation, as a linear combination of another set of covariates: lnα_j = z_jγ.

We will assume that the number of deaths is a function of age, whereas the lnα 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:

<table>
<thead>
<tr>
<th>Iteration</th>
<th>log likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-187.067 (not concave)</td>
</tr>
<tr>
<td>1</td>
<td>-138.13047</td>
</tr>
<tr>
<td>2</td>
<td>-131.83164</td>
</tr>
<tr>
<td>3</td>
<td>-131.59551</td>
</tr>
<tr>
<td>4</td>
<td>-131.5795</td>
</tr>
<tr>
<td>5</td>
<td>-131.57948</td>
</tr>
<tr>
<td>6</td>
<td>-131.57948</td>
</tr>
</tbody>
</table>

Fitting full model:

<table>
<thead>
<tr>
<th>Iteration</th>
<th>log likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-124.34327</td>
</tr>
<tr>
<td>1</td>
<td>-117.76701</td>
</tr>
<tr>
<td>2</td>
<td>-117.56403</td>
</tr>
<tr>
<td>3</td>
<td>-117.56164</td>
</tr>
<tr>
<td>4</td>
<td>-117.56164</td>
</tr>
</tbody>
</table>

Generalized negative binomial regression

<table>
<thead>
<tr>
<th>Number of obs</th>
<th>LR chi2(1)</th>
<th>Prob &gt; chi2</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>28.04</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Log likelihood = -117.56164

| deaths       | Coef.   | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|--------------|---------|-----------|-------|------|---------------------|
| deaths       |         |           |       |      |                     |
| age_mos      | -.0516657 | .0051747 | -9.98 | 0.000 | -.061808 to -.0415233 |
| _cons        | -1.867225 | .2227944 | -8.38 | 0.000 | -2.303894 to -1.430556 |
| logexp       | 1        |           |       |      | (offset)            |

| lnalpha      |         |           |       |      |                     |
| cohort       |         |           |       |      |                     |
| 1960-1967    | .0939546 | .7187747  | 0.13  | 0.896 | -1.314818 to 1.502727 |
| 1968-1976    | .0815279 | .7365476  | 0.11  | 0.912 | -1.362079 to 1.525135 |
| _cons        | -.4759581 | .5156502 | -0.92 | 0.356 | -1.486614 to .5346978 |

We find that age is a significant determinant of the number of deaths. The standard errors for the variables in the lnα 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 $\alpha$ 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-$R^2$-squared
- `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)`: $\chi^2$
- `e(chi2_c)`: $\chi^2$ 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 $\chi^2$ test
- `e(chi2_ct)`: Wald or LR; type of model $\chi^2$ 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(ml_method)`: `max` or `min`; whether optimizer is to perform maximization or minimization
- `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

`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_eq_model)` number of dependent variables
- `e(df_m)` model degrees of freedom
- `e(r2_p)` pseudo-$R$-squared
- `e(ll)` log likelihood
- `e(ll_0)` log likelihood, constant-only model
- `e(N_cluster)` number of clusters
- `e(chi2)` \( \chi^2 \)
- `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 \( \chi^2 \) 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
Methods and formulas

See \texttt{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:

\textit{Mean-dispersion model}

\textit{Constant-dispersion model}

\textbf{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

\[
f(y_j | \nu_j) = \frac{(\nu_j \mu_j)^{y_j} e^{-\nu_j \mu_j}}{\Gamma(y_j + 1)}
\]

where \( \mu_j = \exp(x_j \beta + \text{offset}_j) \) and \( \nu_j \) 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 \( \alpha \), where \( \alpha \) is our ancillary parameter.

The unconditional likelihood for the \( j \)th observation is therefore

\[
f(y_j) = \int_{0}^{\infty} f(y_j | \nu) g(\nu) \, d\nu = \frac{\Gamma(m + y_j)}{\Gamma(y_j + 1) \Gamma(m)} p_j^m (1 - p_j)^{y_j}
\]

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

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

\[
\ln L = \sum_{j=1}^{n} w_j \left[ \ln \{ \Gamma(m + y_j) \} - \ln \{ \Gamma(y_j + 1) \} - \ln \{ \Gamma(m) \} + m \ln(p_j) + y_j \ln(1 - p_j) \right]
\]

For \texttt{gnbreg}, \( \alpha \) can vary across the observations according to the parameterization \( \ln \alpha_j = z_j \gamma \).

\textbf{Constant-dispersion model}

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

\[
m_j = \mu_j/\delta \quad p = 1/(1 + \delta)
\]
\[ \ln L = \sum_{j=1}^{n} w_j \left[ \ln \{ \Gamma(m_j + y_j) \} - \ln \{ \Gamma(y_j + 1) \} \\
- \ln \{ \Gamma(m_j) \} + m_j \ln(p) + y_j \ln(1 - p) \right] \]

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

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] **znb** — 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**