**Description**

`xtnbreg` fits random-effects and conditional fixed-effects overdispersion models where the random effects or fixed effects apply to the distribution of the dispersion parameter. The dispersion is the same for all observations in the same panel. In the random-effects model, the dispersion varies randomly from group to group, such that the inverse of one plus the dispersion follows a Beta distribution. In the fixed-effects model, the dispersion parameter in a group can take on any value.

`xtnbreg` also fits a population-averaged negative binomial model for a nonnegative count dependent variable with overdispersion.

**Quick start**

Random-effects negative-binomial regression of y on x and indicators for levels of categorical variable a using `xtset` data

```
 xtnbreg y x i.a
```

As above, but report incidence-rate ratios

```
 xtnbreg y x i.a, irr
```

Conditional fixed-effects model with exposure variable `evar`

```
 xtnbreg y x i.a, fe exposure(evar)
```

Population-averaged model with robust standard errors

```
 xtnbreg y x i.a, pa vce(robust)
```

**Menu**

Statistics > Longitudinal/panel data > Count outcomes > Negative binomial regression (FE, RE, PA)
Syntax

Random-effects (RE) and conditional fixed-effects (FE) overdispersion models

```
xtnbreg depvar [indepvars] [if] [in] [weight] [, re|fe] RE/FE_options
```

Population-averaged (PA) model

```
xtnbreg depvar [indepvars] [if] [in] [weight], pa [PA_options]
```

<table>
<thead>
<tr>
<th>RE/FE_options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td></td>
</tr>
<tr>
<td><code>noconstant</code></td>
<td>suppress constant term; not available with fe</td>
</tr>
<tr>
<td><code>re</code></td>
<td>use random-effects estimator; the default</td>
</tr>
<tr>
<td><code>fe</code></td>
<td>use fixed-effects estimator</td>
</tr>
<tr>
<td><code>exposure(varname)</code></td>
<td>include ln(varname) in model with coefficient constrained to 1</td>
</tr>
<tr>
<td><code>offset(varname)</code></td>
<td>include varname in model with coefficient constrained to 1</td>
</tr>
<tr>
<td><code>constraints(constraints)</code></td>
<td>apply specified linear constraints</td>
</tr>
<tr>
<td><code>vce(vcetype)</code></td>
<td>vcetype may be oim, bootstrap, or jackknife</td>
</tr>
<tr>
<td><strong>SE</strong></td>
<td></td>
</tr>
<tr>
<td><code>level(#)</code></td>
<td>set confidence level; default is level(95)</td>
</tr>
<tr>
<td><code>irr</code></td>
<td>report incidence-rate ratios</td>
</tr>
<tr>
<td><code>lrmodel</code></td>
<td>perform the likelihood-ratio model test instead of the default Wald test</td>
</tr>
<tr>
<td><code>nocnsreport</code></td>
<td>do not display constraints</td>
</tr>
<tr>
<td><code>display_options</code></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>
<tr>
<td><strong>Maximization</strong></td>
<td></td>
</tr>
<tr>
<td><code>maximize_options</code></td>
<td>control the maximization process; seldom used</td>
</tr>
<tr>
<td><code>collinear</code></td>
<td>keep collinear variables</td>
</tr>
<tr>
<td><code>coeflegend</code></td>
<td>display legend instead of statistics</td>
</tr>
<tr>
<td><strong>PA</strong>_options</td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>

**Model**

- **noconstant**
- **pa**
- **exposure(varname)**
- **offset(varname)**

**Correlation**

- **corr(correlation)**
- **force**

**SE/Robust**

- **vce(vcetype)**
- **nmp**
- **scale(parm)**

**Reporting**

- **level(#)**
- **irr**
- **display_options**

**Optimization**

- **optimize_options**
- **coeflegend**

**correlation**

- **exchangeable**
- **independent**
- **unstructured**
- **fixed matname**
- **ar #**
- **stationary #**
- **nonstationary #**

A panel variable must be specified. For **xtnbreg**, **pa**, correlation structures other than **exchangeable** and **independent** require that a time variable also be specified. Use **xtset**; see [XT] **xtset**.

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

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

by, **mi estimate**, and **statsby** are allowed; see [U] 11.1.10 Prefix commands. **fp** is allowed for the random-effects and fixed-effects models.

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

**iweights**, **fweights**, and **pweights** are allowed for the population-averaged model, and **iweights** are allowed in the random-effects and fixed-effects models; see [U] 11.1.6 **weight**. Weights must be constant within panel.

**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 RE/FE models

noconstant; see [R] Estimation options.

re requests the random-effects estimator, which is the default.
fe requests the conditional fixed-effects estimator.

exposure(varname), offset(varname), 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) and that use bootstrap or jackknife methods (bootstrap, jackknife); see [XT] vce_options.

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

iivr reports exponentiated coefficients e^b rather than coefficients b. For the negative binomial model, exponentiated coefficients have the interpretation of incidence-rate ratios.

lrmodel, noconsreport; 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(#), [nolog, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(init_specs)]; see [R] Maximize. These options are seldom used.

The following options are available with xtnbreg but are not shown in the dialog box:
collinear, coeflegend; see [R] Estimation options.

Options for PA model

noconstant; see [R] Estimation options.

pa requests the population-averaged estimator.

exposure(varname), offset(varname); see [R] Estimation options.

corr(correlation) specifies the within-panel correlation structure; the default corresponds to the equal-correlation model, corr(exchangeable).

When you specify a correlation structure that requires a lag, you indicate the lag after the structure’s name with or without a blank; for example, corr(ar 1) or corr(ar1).
If you specify the fixed correlation structure, you specify the name of the matrix containing the assumed correlations following the word *fixed*, for example, `corr(fixed myr)`.

`force` specifies that estimation be forced even though the time variable is not equally spaced. This is relevant only for correlation structures that require knowledge of the time variable. These correlation structures require that observations be equally spaced so that calculations based on lags correspond to a constant time change. If you specify a time variable indicating that observations are not equally spaced, the (time dependent) model will not be fit. If you also specify `force`, the model will be fit, and it will be assumed that the lags based on the data ordered by the time variable are appropriate.

**vce(vcetype)** specifies the type of standard error reported, which includes types that are derived from asymptotic theory (*conventional*), that are robust to some kinds of misspecification (*robust*), and that use bootstrap or jackknife methods (*bootstrap*, *jackknife*); see [XT] *vce_options*.

`vce(conventional)`, the default, uses the conventionally derived variance estimator for generalized least-squares regression.

`nmp, scale(x2 | dev | phi | #)`; see [XT] *vce_options*.

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

`irr` reports exponentiated coefficients $e^b$ rather than coefficients $b$. For the negative binomial model, exponentiated coefficients have the interpretation of incidence-rate ratios.

**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*.

**Optimize** options control the iterative optimization process. These options are seldom used.

`iterate(#)` specifies the maximum number of iterations. When the number of iterations equals #, the optimization stops and presents the current results, even if convergence has not been reached. The default is `iterate(100)`.

`tolerance(#)` specifies the tolerance for the coefficient vector. When the relative change in the coefficient vector from one iteration to the next is less than or equal to #, the optimization process is stopped. `tolerance(1e-6)` is the default.

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [R] *set iter*.

`trace` specifies that the current estimates be printed at each iteration.

The following option is available with *xtnbreg* but is not shown in the dialog box: `coeflegend`; see [R] *Estimation options*.
xtnbreg fits random-effects overdispersion models, conditional fixed-effects overdispersion models, and population-averaged negative binomial models. Here “random effects” and “fixed effects” apply to the distribution of the dispersion parameter, not to the $\alpha/\beta$ term in the model. In the random-effects and fixed-effects overdispersion models, the dispersion is the same for all elements in the same group (that is, elements with the same value of the panel variable). In the random-effects model, the dispersion varies randomly from group to group, such that the inverse of one plus the dispersion follows a Beta($r, s$) distribution. In the fixed-effects model, the dispersion parameter in a group can take on any value, because a conditional likelihood is used in which the dispersion parameter drops out of the estimation.

By default, the population-averaged model is an equal-correlation model; xtnbreg, pa assumes corr(exchangeable). Thus, xtnbreg is a convenience command for fitting the population-averaged using xtgee; see [XT] xtgee. Typing

```
.xtnbreg ..., ..., pa exposure(time)
```

is equivalent to typing

```
.xtgee ..., ..., family(nbinomial) link(log) corr(exchangeable) exposure(time)
```

See also [XT] xtgee for information about xtnbreg.

By default, or when re is specified, xtnbreg fits a maximum-likelihood random-effects overdispersion model.

> Example 1

You have (fictional) data on injury “incidents” incurred among 20 airlines in each of 4 years. (Incidents range from major injuries to exceedingly minor ones.) The government agency in charge of regulating airlines has run an experimental safety training program, and, in each of the years, some airlines have participated and some have not. You now wish to analyze whether the “incident” rate is affected by the program. You choose to estimate using random-effects negative binomial regression, as the dispersion might vary across the airlines for unidentified airline-specific reasons. Your measure of exposure is passenger miles for each airline in each year.
. use https://www.stata-press.com/data/r16/airacc
. xtnbreg i_cnt inprog, exposure(pmiles) irr

Fitting negative binomial (constant dispersion) model:
Iteration 0: log likelihood = -293.57997
Iteration 1: log likelihood = -293.57997
(output omitted)

Fitting full model:
Iteration 0: log likelihood = -295.72633
Iteration 1: log likelihood = -270.49929 (not concave)
(output omitted)

Random-effects negative binomial regression
Number of obs = 80
Group variable: airline
Number of groups = 20
Random effects u_i ~ Beta
Obs per group:
min = 4
avg = 4.0
max = 4
Wald chi2(1) = 2.04
Log likelihood = -265.38202

|      | IRR    | Std. Err. | z     | P>|z|  | [95% Conf. Interval] |
|------|--------|-----------|-------|------|---------------------|
| i_cnt|        |           |       |      |                     |
|      |        |           |       |      |                     |
| inprog | .911673 | .0590277 | -1.43 | 0.153 | .8030206 1.035027 |
| _cons | .0367524 | .0407032 | -2.98 | 0.003 | .0041936 .3220983 |
| ln(pmiles) | 1 (exposure) | | | | |
| /ln_r | 4.794991 | .951781 | | | 2.929535 6.660448 |
| /ln_s | 3.268052 | .4709033 | | | 2.345098 4.191005 |
| r | 120.9033 | 115.0735 | | 18.71892 | 780.9007 |
| s | 26.26013 | 12.36598 | | 10.4343 | 66.08918 |

Note: Estimates are transformed only in the first equation.
Note: _cons estimates baseline incidence rate (conditional on zero random effects).

LR test vs. pooled: chibar2(01) = 19.03 Prob >= chibar2 = 0.000

In the output above, the /ln_r and /ln_s lines refer to ln(r) and ln(s), where the inverse of one plus the dispersion is assumed to follow a Beta(r, s) distribution. The output also includes a likelihood-ratio test, which compares the panel estimator with the pooled estimator (that is, a negative binomial estimator with constant dispersion).

You find that the incidence rate for accidents is not significantly different for participation in the program and that the panel estimator is significantly different from the pooled estimator.
We may alternatively fit a fixed-effects overdispersion model:

```
xtnbreg i_cnt inprog, exposure(pmiles) irr fe nolog
```

Conditional FE negative binomial regression
Number of obs = 80
Group variable: airline
Number of groups = 20

<table>
<thead>
<tr>
<th>Obs per group:</th>
</tr>
</thead>
<tbody>
<tr>
<td>min = 4</td>
</tr>
<tr>
<td>avg = 4.0</td>
</tr>
<tr>
<td>max = 4</td>
</tr>
</tbody>
</table>

Wald chi2(1) = 2.11
Log likelihood = -174.25143
Prob > chi2 = 0.1463

| i_cnt | IRR   | Std. Err. | z     | P>|z|  | [95% Conf. Interval] |
|-------|-------|-----------|-------|------|----------------------|
| inprog| .9062669 | .0613917  | -1.45 | 0.146 | .793587  1.034946    |
| _cons | .0329025 | .0331262  | -3.39 | 0.001 | .0045734 .2367111  |

Note: _cons estimates baseline incidence rate (conditional on zero random effects).

Example 2

We rerun our previous example, but this time we fit a robust equal-correlation population-averaged model:

```
xtnbreg i_cnt inprog, exposure(pmiles) irr vce(robust) pa
```

Iteration 1: tolerance = .02499392
Iteration 2: tolerance = .0000482
Iteration 3: tolerance = 2.929e-07

GEE population-averaged model
Number of obs = 80
Group variable: airline
Number of groups = 20
Link: log
Family: negative binomial(k=1)
Correlation: exchangeable

<table>
<thead>
<tr>
<th>Obs per group:</th>
</tr>
</thead>
<tbody>
<tr>
<td>min = 4</td>
</tr>
<tr>
<td>avg = 4.0</td>
</tr>
<tr>
<td>max = 4</td>
</tr>
</tbody>
</table>

Wald chi2(1) = 1.28
Scale parameter: 1
(Std. Err. adjusted for clustering on airline)

| i_cnt | Semirobust IRR | Std. Err. | z     | P>|z|  | [95% Conf. Interval] |
|-------|----------------|-----------|-------|------|----------------------|
| inprog| .927275       | .0617857  | -1.13 | 0.257| .8137513  1.056636  |
| _cons| .0080211      | .0004117  | -94.02| 0.000| .0072535  .00887  |

ln(pmiles) 1 (exposure)

Note: _cons estimates baseline incidence rate (conditional on zero random effects).
We compare this with a pooled estimator with clustered robust-variance estimates:

```
. nbreg i_cnt inprog, exposure(pmiles) irr vce(cluster airline)
```

Fitting Poisson model:

```
Iteration 0:  log pseudolikelihood = -293.57997
Iteration 1:  log pseudolikelihood = -293.57997
```

Fitting constant-only model:

```
Iteration 0:  log pseudolikelihood = -335.13615
Iteration 1:  log pseudolikelihood = -279.43327
Iteration 2:  log pseudolikelihood = -276.09296
Iteration 3:  log pseudolikelihood = -274.84036
Iteration 4:  log pseudolikelihood = -274.81076
Iteration 5:  log pseudolikelihood = -274.81075
```

Fitting full model:

```
Iteration 0:  log pseudolikelihood = -274.56985
Iteration 1:  log pseudolikelihood = -274.55077
Iteration 2:  log pseudolikelihood = -274.55077
```

Negative binomial regression

<table>
<thead>
<tr>
<th></th>
<th>Number of obs =</th>
<th>Wald chi2(1) =</th>
<th>Prob &gt; chi2 =</th>
<th>Pseudo R2 =</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispersion</td>
<td>mean</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log pseudolikelihood</td>
<td>-274.55077</td>
<td></td>
<td>0.60</td>
<td>0.4369</td>
</tr>
</tbody>
</table>

(Std. Err. adjusted for 20 clusters in airline)

<table>
<thead>
<tr>
<th></th>
<th>Robust</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i_cnt</td>
<td>IRR Std. Err.</td>
<td>z P&gt;</td>
</tr>
<tr>
<td>inprog</td>
<td>.9429015</td>
<td>.0713091 -0.78 0.437</td>
</tr>
<tr>
<td>_cons</td>
<td>.007956</td>
<td>.0004237 -90.77 0.000</td>
</tr>
<tr>
<td>ln(pmiles)</td>
<td>1 (exposure)</td>
<td></td>
</tr>
<tr>
<td>/lnalpha</td>
<td>-2.835089</td>
<td>.3351784 -3.492027 -2.178151</td>
</tr>
<tr>
<td>alpha</td>
<td>.0587133</td>
<td>.0196794 .0304391 .1132507</td>
</tr>
</tbody>
</table>

Note: Estimates are transformed only in the first equation.
Note: _cons estimates baseline incidence rate.
Stored results

`xtnbreg, re` stores the following in `e()`:

Scalars
- `e(N)` number of observations
- `e(N_g)` number of groups
- `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(ll)` log likelihood
- `e(ll_0)` log likelihood, constant-only model
- `e(ll_c)` log likelihood, comparison model
- `e(chi2)` \( \chi^2 \)
- `e(chi2_c)` \( \chi^2 \) for comparison test
- `e(chi2_min)` smallest group size
- `e(chi2_avg)` average group size
- `e(chi2_max)` largest group size
- `e(r)` value of \( r \) in Beta(\( r \), \( s \))
- `e(s)` value of \( s \) in Beta(\( r \), \( s \))
- `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)` `xtnbreg`
- `e(cmdline)` command as typed
- `e(depvar)` name of dependent variable
- `e(ivar)` variable denoting groups
- `e(model)` `re`
- `e(wtype)` weight type
- `e(wexp)` weight expression
- `e(title)` title in estimation output
- `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(vce)` vcetype specified in `vce()`
- `e(method)` estimation method
- `e(distrib)` Beta; the distribution of the random effect
- `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
- `e(gradient)` gradient vector
- `e(V)` variance–covariance matrix of the estimators

Functions
- `e(sample)` marks estimation sample
xtnbreg, fe stores the following in e():

Scalars

- `e(N)`: number of observations
- `e(N_g)`: number of groups
- `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(ll)`: log likelihood
- `e(ll_0)`: log likelihood, constant-only model
- `e(chi2)`: $\chi^2$ test
- `e(g_min)`: smallest group size
- `e(g_avg)`: average group size
- `e(g_max)`: largest group size
- `e(p)`: p-value for model test
- `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)`: `xtnbreg`
- `e(cmdline)`: command as typed
- `e(depvar)`: name of dependent variable
- `e(ivar)`: variable denoting groups
- `e(model)`: `fe`
- `e(wtype)`: weight type
- `e(wexp)`: weight expression
- `e(title)`: title in estimation output
- `e(offset)`: linear offset variable
- `e(chi2type)`: LR; type of model $\chi^2$ test
- `e(vce)`: vcetype specified in `vce()`
- `e(method)`: requested estimation method
- `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
- `e(gradient)`: gradient vector
- `e(V)`: variance–covariance matrix of the estimators

Functions

- `e(sample)` marks estimation sample
xtnbreg, pa stores the following in e():

Scalars
- e(N): number of observations
- e(N_g): number of groups
- e(df_m): model degrees of freedom
- e(chi2): $\chi^2$
- e(p): $p$-value for model test
- e(df_pear): degrees of freedom for Pearson $\chi^2$
- e(chi2_dev): $\chi^2$ test of deviance
- e(chi2_dis): $\chi^2$ test of deviance dispersion
- e(deviance): deviance
- e(dispers): deviance dispersion
- e(phi): scale parameter
- e(g_min): smallest group size
- e(g_avg): average group size
- e(g_max): largest group size
- e(rank): rank of e(V)
- e(tol): target tolerance
- e(dif): achieved tolerance
- e(rc): return code

Macros
- e(cmd): xtgee
- e(cmd2): xtnbreg
- e(cmdline): command as typed
- e(depvar): name of dependent variable
- e(ivar): variable denoting groups
- e(tvar): variable denoting time within groups
- e(model): pa
- e(family): negative binomial($k=1$)
- e(link): log; link function
- e(corr): correlation structure
- e(scale): x2, dev, phi, or #; scale parameter
- e(type): weight type
- e(wexp): weight expression
- e(offset): linear offset variable
- e(chi2type): Wald; type of model $\chi^2$ test
- e(vce): vcetype specified in vce()
- e(vcetype): title used to label Std. Err.
- e(nmp): nmp, if specified
- e(nbalpha): $\alpha$
- e(properties): b V
- e(predict): program used to implement predict
- 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(R): estimated working correlation matrix
- e(V): variance–covariance matrix of the estimators
- e(V_modelbased): model-based variance

Functions
- e(sample): marks estimation sample
Methods and formulas

xtnbreg, pa reports the population-averaged results obtained by using xtgee, family(nbinomial) link(log) to obtain estimates. See [XT] xtgee for details on the methods and formulas.

For the random-effects and fixed-effects overdispersion models, let $y_{it}$ be the count for the $i$th observation in the $t$th group. We begin with the model

$$y_{it} | \gamma_{it} \sim \text{Poisson}(\gamma_{it}),$$

where $\gamma_{it} | \delta_i \sim \text{gamma}(\lambda_{it}, \delta_i)$ with $\lambda_{it} = \exp(x_{it}\beta + \text{offset}_{it})$ and $\delta_i$ is the dispersion parameter. This yields the model

$$\Pr(Y_{it} = y_{it} | x_{it}, \delta_i) = \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it} + 1)} \left( \frac{1}{1 + \delta_i} \right)^{\lambda_{it}} \left( \frac{\delta_i}{1 + \delta_i} \right)^{y_{it}}$$

(See Hausman, Hall, and Griliches [1984, eq. 3.1, 922]; our $\delta$ is the inverse of their $\delta$.) Looking at within-panel effects only, we find that this specification yields a negative binomial model for the $i$th group with dispersion (variance divided by the mean) equal to $1 + \delta_i$, that is, constant dispersion within group. This parameterization of the negative binomial model differs from the default parameterization of nbreg, which has dispersion equal to $1 + \alpha \exp(x_{it}\beta + \text{offset})$; see [R] nbreg.

For a random-effects overdispersion model, we allow $\delta_i$ to vary randomly across groups; namely, we assume that $1/(1 + \delta_i) \sim \text{Beta}(r, s)$. The joint probability of the counts for the $i$th group is

$$\Pr(Y_{i1} = y_{i1}, \ldots, Y_{in_i} = y_{in_i} | X_i) = \int_0^\infty \prod_{t=1}^{n_i} \Pr(Y_{it} = y_{it} | x_{it}, \delta_i) f(\delta_i) d\delta_i$$

$$= \frac{\Gamma(r+s)\Gamma(r+\sum_{t=1}^{n_i} \lambda_{it})\Gamma(s+\sum_{t=1}^{n_i} y_{it})}{\Gamma(r)\Gamma(s)\Gamma(r+s+\sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it})} \prod_{t=1}^{n_i} \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it} + 1)}$$

for $X_i = (x_{i1}, \ldots, x_{in_i})$ and where $f$ is the probability density function for $\delta_i$. The resulting log likelihood is

$$\ln L = \sum_{i=1}^{n} w_i \left[ \ln \Gamma(r+s) + \ln \Gamma(r + \sum_{k=1}^{n_i} \lambda_{ik}) + \ln \Gamma(s + \sum_{k=1}^{n_i} y_{ik}) - \ln \Gamma(r) - \ln \Gamma(s) \right.$$

$$\left. - \ln \Gamma(r + s + \sum_{k=1}^{n_i} \lambda_{ik} + \sum_{k=1}^{n_i} y_{ik}) + \sum_{t=1}^{n_i} \left\{ \ln \Gamma(\lambda_{it} + y_{it}) - \ln \Gamma(\lambda_{it}) - \ln \Gamma(y_{it} + 1) \right\} \right]$$

where $\lambda_{it} = \exp(x_{it}\beta + \text{offset}_{it})$ and $w_i$ is the weight for the $i$th group (Hausman, Hall, and Griliches 1984, eq. 3.5, 927).

For the fixed-effects overdispersion model, we condition the joint probability of the counts for each group on the sum of the counts for the group (that is, the observed $\sum_{t=1}^{n_i} y_{it}$). This yields

$$\Pr(Y_{i1} = y_{i1}, \ldots, Y_{in_i} = y_{in_i} | X_i, \sum_{t=1}^{n_i} Y_{it} = \sum_{t=1}^{n_i} y_{it})$$

$$= \frac{\Gamma(\sum_{t=1}^{n_i} \lambda_{it})\Gamma(\sum_{t=1}^{n_i} y_{it} + 1)}{\Gamma(\sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it})} \prod_{t=1}^{n_i} \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it} + 1)}$$
The conditional log likelihood is

\[
\ln L = \sum_{i=1}^{n} w_i \left[ \ln \left( \sum_{t=1}^{n_i} \lambda_{it} \right) + \ln \left( \sum_{t=1}^{n_i} y_{it} + 1 \right) - \ln \left( \sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it} \right) \right] \\
+ \sum_{t=1}^{n_i} \left\{ \ln \Gamma(\lambda_{it} + y_{it}) - \ln \Gamma(\lambda_{it}) - \ln \Gamma(y_{it} + 1) \right\}
\]

See Hausman, Hall, and Griliches (1984) for a more thorough development of the random-effects and fixed-effects models. Also see Cameron and Trivedi (2013) for a good textbook treatment of this model.

References


Also see

[XT] xtnbreg postestimation — Postestimation tools for xtnbreg

[XT] xtgee — Fit population-averaged panel-data models by using GEE

[XT] xtpoisson — Fixed-effects, random-effects, and population-averaged Poisson models

[XT] xtset — Declare data to be panel data

[ME] membreg — Multilevel mixed-effects negative binomial regression

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

[R] nbreg — Negative binomial regression

[U] 20 Estimation and postestimation commands