xtnbreg — Fixed-effects, random-effects, & population-averaged negative binomial models

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
```

Same 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*]

RE/FE_options	Description
Model	
`noconstant`	suppress constant term; not available with `fe`
`re`	use random-effects estimator; the default
`fe`	use fixed-effects estimator
`exposure(`*varname*`)`	include ln(*varname*) in model with coefficient constrained to 1
`offset(`*varname*`)`	include *varname* in model with coefficient constrained to 1
`constraints(`*constraints*`)`	apply specified linear constraints
SE	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`irr`	report incidence-rate ratios
`lrmodel`	perform the likelihood-ratio model test instead of the default Wald test
`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

PA_options	Description
Model	
noconstant	suppress constant term
pa	use population-averaged estimator
exposure(*varname*)	include ln(*varname*) in model with coefficient constrained to 1
offset(*varname*)	include *varname* in model with coefficient constrained to 1
Correlation	
corr(*correlation*)	within-panel correlation structure
force	estimate even if observations unequally spaced in time
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, bootstrap, or jackknife
nmp	use divisor $N - P$ instead of the default N
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
irr	report incidence-rate ratios
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Optimization	
optimize_options	control the optimization process; seldom used
coeflegend	display legend instead of statistics

correlation	Description
exchangeable	exchangeable
independent	independent
unstructured	unstructured
fixed *matname*	user-specified
ar #	autoregressive of order #
stationary #	stationary of order #
nonstationary #	nonstationary of order #

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**.

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

by, collect, mi estimate, and statsby are allowed; see [U] **11.1.10 Prefix commands**. bayes is allowed for the random-effects model. For more details, see [BAYES] **bayes: xtnbreg**. 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

Model

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**.

SE

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***.

Reporting

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.

lrmodel, 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.

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

Model

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

pa requests the population-averaged estimator.

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

Correlation

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.

SE/Robust

`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***.

Reporting

`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**.

Optimization

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**.

Remarks and examples

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 $\mathbf{x}\boldsymbol{\beta}$ 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/r19/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                          Prob > chi2       = 0.1532

------------------------------------------------------------------------------
       i_cnt |        IRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      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 to incidence-rate
      ratios.
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
                                                     Obs per group:
                                                                  min =      4
                                                                  avg =    4.0
                                                                  max =      4
                                                     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
  ln(pmiles) |          1  (exposure)
------------------------------------------------------------------------------
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
Family: Negative binomial(k=1)                       Obs per group:
Link:   Log                                                       min =      4
Correlation: exchangeable                                         avg =    4.0
                                                                  max =      4
                                                     Wald chi2(1)     =   1.28
Scale parameter = 1                                  Prob > chi2      = 0.2571
                                (Std. err. adjusted for clustering on airline)
------------------------------------------------------------------------------
             |             Semirobust
       i_cnt |        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                        Number of obs =     80
                                                    Wald chi2(1)  =   0.60
Dispersion: mean                                    Prob > chi2   = 0.4369
Log pseudolikelihood = -274.55077                   Pseudo R2     = 0.0009
                              (Std. err. adjusted for 20 clusters in airline)
------------------------------------------------------------------------------
             |               Robust
       i_cnt |        IRR   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
      inprog |   .9429015   .0713091    -0.78   0.437     .8130032    1.093555
       _cons |    .007956   .0004237   -90.77   0.000     .0071674    .0088314
  ln(pmiles) |          1  (exposure)
-------------+----------------------------------------------------------------
    /lnalpha |  -2.835089   .3351784                     -3.492027   -2.178151
-------------+----------------------------------------------------------------
       alpha |   .0587133   .0196794                      .0304391    .1132507
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation to incidence-rate
      ratios.
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)	χ^2
e(chi2_c)	χ^2 for comparison test
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_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 χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^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

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.

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)	χ^2
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 χ^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

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.

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)	χ^2
e(p)	p-value for model test
e(df_pear)	degrees of freedom for Pearson χ^2
e(chi2_dev)	χ^2 test of deviance
e(chi2_dis)	χ^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(wtype)	weight type
e(wexp)	weight expression
e(offset)	linear offset variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(nmp)	nmp, if specified
e(nbalpha)	α
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

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

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 tth observation in the ith 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 $\lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it})$ and δ_i is the dispersion parameter. This yields the model

$$\Pr(Y_{it} = y_{it} \mid \mathbf{x}_{it}, \delta_i) = \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)} \left(\frac{1}{1+\delta_i}\right)^{\lambda_{it}} \left(\frac{\delta_i}{1+\delta_i}\right)^{y_{it}}$$

(See Hausman, Hall, and Griliches [1984, eq. 3.1, 922]; our δ is the inverse of their δ.) Looking at within-panel effects only, we find that this specification yields a negative binomial model for the ith 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 $1 + \alpha \exp(\mathbf{x}\boldsymbol{\beta} + \text{offset})$; see [R] **nbreg**.

For a random-effects overdispersion model, we allow δ_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 ith group is

$$\begin{aligned}\Pr(Y_{i1} = y_{i1}, \ldots, Y_{in_i} = y_{in_i} | \mathbf{X}_i) &= \int_0^\infty \prod_{t=1}^{n_i} \Pr(Y_{it} = y_{it} \mid \mathbf{x}_{it}, \delta_i)\, f(\delta_i)\, d\delta_i \\ &= \frac{\Gamma(r+s)\Gamma(r + \sum_{t=1}^{n_i}\lambda_{it})\Gamma(s + \sum_{t=1}^{n_i} y_{it})}{\Gamma(r)\Gamma(s)\Gamma(r + s + \sum_{t=1}^{n_i}\lambda_{it} + \sum_{t=1}^{n_i} y_{it})} \prod_{t=1}^{n_i} \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)}\end{aligned}$$

for $\mathbf{X}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i})$ and where f is the probability density function for δ_i. The resulting log likelihood is

$$\begin{aligned}\ln L = \sum_{i=1}^{n} w_i \Bigg[& \ln\Gamma(r+s) + \ln\Gamma\left(r + \sum_{k=1}^{n_i} \lambda_{ik}\right) + \ln\Gamma\left(s + \sum_{k=1}^{n_i} y_{ik}\right) - \ln\Gamma(r) - \ln\Gamma(s) \\ & - \ln\Gamma\left(r + s + \sum_{k=1}^{n_i} \lambda_{ik} + \sum_{k=1}^{n_i} y_{ik}\right) + \sum_{t=1}^{n_i} \Big\{ \ln\Gamma(\lambda_{it} + y_{it}) - \ln\Gamma(\lambda_{it}) - \ln\Gamma(y_{it}+1) \Big\} \Bigg]\end{aligned}$$

where $\lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta} + \text{offset}_{it})$ and w_i is the weight for the ith 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

$$\begin{aligned}\Pr\Big(Y_{i1} = y_{i1}, \ldots, Y_{in_i} = y_{in_i} \;\Big|\; & \mathbf{X}_i, \textstyle\sum_{t=1}^{n_i} Y_{it} = \sum_{t=1}^{n_i} y_{it}\Big) \\ & = \frac{\Gamma(\sum_{t=1}^{n_i} \lambda_{it})\Gamma(\sum_{t=1}^{n_i} y_{it} + 1)}{\Gamma(\sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it})} \prod_{t=1}^{n_i} \frac{\Gamma(\lambda_{it} + y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)}\end{aligned}$$

The conditional log likelihood is

$$\begin{aligned}\ln L = \sum_{i=1}^{n} w_i \Bigg[& \ln\Gamma\left(\sum_{t=1}^{n_i} \lambda_{it}\right) + \ln\Gamma\left(\sum_{t=1}^{n_i} y_{it} + 1\right) - \ln\Gamma\left(\sum_{t=1}^{n_i} \lambda_{it} + \sum_{t=1}^{n_i} y_{it}\right) \\ & + \sum_{t=1}^{n_i} \Big\{ \ln\Gamma(\lambda_{it} + y_{it}) - \ln\Gamma(\lambda_{it}) - \ln\Gamma(y_{it}+1) \Big\} \Bigg]\end{aligned}$$

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

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

Guimarães, P. 2005. A simple approach to fit the beta-binomial model. *Stata Journal* 5: 385–394.

Hausman, J. A., B. H. Hall, and Z. Griliches. 1984. Econometric models for count data with an application to the patents–R & D relationship. *Econometrica* 52: 909–938. https://doi.org/10.2307/1911191.

Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22. https://doi.org/10.1093/biomet/73.1.13.

Also see

[XT] **xtnbreg postestimation** — Postestimation tools for xtnbreg

[XT] **xtgee** — GEE population-averaged panel-data models

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

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtnbreg** — Bayesian random-effects negative binomial model

[ME] **menbreg** — 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

For suggested citations, see the FAQ on citing Stata documentation.