Title

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

Syntax

Random-effects and conditional fixed-effects overdispersion models

`xtnbreg` *depvar* [*varlist*] [*weight*] [`if` *exp*] [`in` *range*] [, { `re` | `fe` } `i(`*varname*`)` `irr` `noconstant` `noskip` `exposure(`*varname*`)` `offset(`*varname*`)` `level(`*#*`)` *maximize_options*]

Population-averaged model

`xtnbreg` *depvar* [*varlist*] [*weight*] [`if` *exp*] [`in` *range*] , `pa` [`i(`*varname*`)` `irr` `robust` `noconstant` `exposure(`*varname*`)` `offset(`*varname*`)` `level(`*#*`)` *xtgee_options* *maximize_options*]

`iweight`s, `aweight`s, and `pweight`s are allowed for the population-averaged model and `iweight`s are allowed in the random-effects and fixed-effects models; see [U] **14.1.6 weight**. Note that weights must be constant within panels.

`xtnbreg` shares the features of all estimation commands; see [U] **23 Estimation and post-estimation commands**.

Syntax for predict

Random-effects and conditional fixed-effects overdispersion models

`predict` [*type*] *newvarname* [`if` *exp*] [`in` *range*] [, { `xb` | `stdp` } `nooffset`]

Population-averaged model

`predict` [*type*] *newvarname* [`if` *exp*] [`in` *range*] [, { `mu` | `rate` | `xb` | `stdp` } `nooffset`]

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Description

`xtnbreg` estimates 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, and not to the $\mathbf{x}\beta$ 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 (i.e., elements with the same value of the `i()` variable). In the random-effects model, the dispersion varies randomly from group to group such that the inverse of the dispersion has a $\mathrm{Beta}(r, s)$ distribution. In the fixed-effects model, the dispersion parameter in a group can take on any value, since 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` assumes `corr(exchangeable)`. See [R] **xtgee** for details on this option to fit other population-averaged models.

Options

`re` requests the random-effects estimator. `re` is the default if none of `re`, `fe`, and `pa` is specified.

`fe` requests the conditional fixed-effects estimator.

`pa` requests the population-averaged estimator.

`i(`*varname*`)` specifies the variable name that contains the unit to which the observation belongs. You can specify the `i()` option the first time you estimate or use the `iis` command to set `i()` beforehand. After that, Stata will remember the variable's identity. See [R] **xt**.

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

`robust` (`pa` only) specifies the Huber/White/sandwich estimator of variance is to be used in place of the IRLS variance estimator; see [R] **xtgee**. This alternative produces valid standard errors even if the correlations within group are not as hypothesized by the specified correlation structure. It does, however, require that the model correctly specifies the mean. As such, the resulting standard errors are labeled "semi-robust" instead of "robust". Note that although there is no `cluster()` option, results are as if there were a `cluster()` option and you specified clustering on `i()`.

`noconstant` suppresses the constant term (intercept) in the model.

`noskip` specifies that a full maximum-likelihood model with only a constant for the regression equation be estimated. This constant-only model is used as the base model to compute a likelihood-ratio χ^2 statistic for the model test. By default, the model test uses an asymptotically equivalent Wald χ^2 statistic. For many models, this option can significantly increase estimation time.

`exposure(`*varname*`)` and `offset(`*varname*`)` are different ways of specifying the same thing. `exposure()` specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname*) with coefficient constrained to be 1 is entered into the regression equation. `offset()` specifies a variable that is to be entered directly into the regression equation with coefficient constrained to be 1; thus exposure is assumed to be $e^{varname}$.

`level(`*#*`)` specifies the confidence level, in percent, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **23.5 Specifying the width of confidence intervals**.

xtgee_options specifies any other options allowed by `xtgee` for `family(nbinom) link(log)`; see [R] **xtgee**.

maximize_options control the maximization process; see [R] **maximize**. Use the `trace` option to view parameter convergence.

Options for predict

`xb` calculates the linear prediction. This is the default for the random-effects and fixed-effects models.

`stdp` calculates the standard error of the linear prediction.

`mu` and `rate` both calculate the predicted probability of *depvar*. `mu` takes into account the `offset()`. `rate` ignores those adjustments. `mu` and `rate` are equivalent if you did not specify `offset()`. `mu` is the default for the population-averaged model.

nooffset is relevant only if you specified **offset(***varname***)** for **xtnbreg**. It modifies the calculations made by **predict** so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_{it}\mathbf{b}$ rather than $\mathbf{x}_{it}\mathbf{b} + \text{offset}_{it}$.

Remarks

xtnbreg is a convenience command if you want the population-averaged model. Typing

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

is equivalent to typing

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

Thus, also see [R] **xtgee** for information about **xtnbreg**.

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

▷ Example

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 because the dispersion might vary across the airlines because of unidentified airline-specific reasons. Your measure of exposure is passenger miles for each airline in each year.

```
. xtnbreg i_cnt inprog, i(airline) exposure(pmiles) irr nolog

Random-effects negative binomial                Number of obs      =        80
Group variable (i) : 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   .0590278     -1.428   0.153       .8030204    1.035027
  pmiles | (exposure)
---------+--------------------------------------------------------------------
   /ln_r |   4.794971    .951754      5.038   0.000       2.929567    6.660374
   /ln_s |   3.268055   .4709027      6.940   0.000       2.345103    4.191007
---------+--------------------------------------------------------------------
       r |   120.9008   115.0679                          18.71953    780.8432
       s |   26.26022   12.36601                          10.43435    66.08933
------------------------------------------------------------------------------
Likelihood ratio test versus pooled: chi2(1) =    18.34   Prob > chi2 = 0.0000
```

In the output above, the **/ln_r** and **/ln_s** lines refer to $\ln(r)$ and $\ln(s)$, where the inverse of the dispersion is assumed to follow a $\text{Beta}(r, s)$ random distribution. The output also includes a likelihood-ratio test which compares the panel estimator with the pooled estimator (i.e., 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 estimate a fixed-effects overdispersion model:

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

Conditional fixed-effects negative binomial     Number of obs      =        80
Group variable (i) : 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 |   .9062668   .0613916     -1.453   0.146       .7935872    1.034946
  pmiles | (exposure)
------------------------------------------------------------------------------
```

◁

▷ Example

Rerunning our previous example in order to fit a robust equal-correlation population-averaged model:

```
. xtnbreg i_cnt inprog, i(airline) exposure(pmiles) eform robust pa

Iteration 1: tolerance = .02488289
Iteration 2: tolerance = .00004846
Iteration 3: tolerance = 2.914e-07

GEE population-averaged model                   Number of obs      =        80
Group variable:                     airline     Number of groups   =        20
Link:                                   log     Obs per group: min =         4
Family:             negative binomial(k=1)                     avg =       4.0
Correlation:                   exchangeable                    max =         4
                                                Wald chi2(1)       =      1.28
Scale parameter:                          1     Prob > chi2        =    0.2578

                          (standard errors adjusted for clustering on airline)
------------------------------------------------------------------------------
         |             Semi-robust
   i_cnt |        IRR   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  inprog |   .9273828   .0617892     -1.131   0.258       .8138524     1.05675
  pmiles | (exposure)
------------------------------------------------------------------------------
```

We may compare this with a pooled estimator with cluster robust variance estimates:

```
. nbreg i_cnt inprog, exposure(pmiles) robust cluster(airline) irr nolog
Negative binomial regression                      Number of obs   =         80
                                                  Wald chi2(1)    =       0.60
Log likelihood = -274.55077                       Prob > chi2     =     0.4369
                          (standard errors adjusted for clustering on airline)
------------------------------------------------------------------------------
         |               Robust
   i_cnt |        IRR   Std. Err.       z     P>|z|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
  inprog |   .9429015   .0713091     -0.777   0.437       .8130031    1.093554
  pmiles | (exposure)
---------+--------------------------------------------------------------------
/lnalpha |  -2.835089   .3351784     -8.458   0.000      -3.492027   -2.178152
---------+--------------------------------------------------------------------
   alpha |   .0587133   .0196794                          .0304391    .1132507
------------------------------------------------------------------------------
```

◁

Saved Results

`xtnbreg, re` saves in `e()`:

Scalars

`e(N)`	number of observations	`e(ll_c)`	log likelihood, comparison model
`e(k)`	number of estimated parameters	`e(df_m)`	model degrees of freedom
`e(k_eq)`	number of equations	`e(chi2)`	model χ^2
`e(k_dv)`	number of dependent variables	`e(p)`	model significance
`e(N_g)`	number of groups	`e(chi2_c)`	χ^2 for comparison test
`e(g_min)`	smallest group size	`e(r)`	value of r in Beta(r,s)
`e(g_avg)`	average group size	`e(s)`	value of s in Beta(r,s)
`e(g_max)`	largest group size	`e(ic)`	number of iterations
`e(ll)`	log likelihood	`e(rc)`	return code
`e(ll_0)`	log likelihood, constant-only model		

Macros

`e(cmd)`	`xtnbreg`	`e(opt)`	type of optimization
`e(cmd2)`	`xtn_re`	`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(depvar)`	name of dependent variable	`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(title)`	title in estimation output		
`e(ivar)`	variable denoting groups	`e(offset)`	offset
`e(wtype)`	weight type	`e(distrib)`	`Beta`; the distribution of the random effect
`e(wexp)`	weight expression		
`e(method)`	estimation method	`e(predict)`	program used to implement `predict`
`e(user)`	name of likelihood-evaluation program		

Matrices

`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

`xtnbreg, fe` saves in `e()`:

Scalars

`e(N)`	number of observations	`e(ll)`	log likelihood
`e(k)`	number of estimated parameters	`e(ll_0)`	log likelihood, constant-only model
`e(k_eq)`	number of equations	`e(df_m)`	model degrees of freedom
`e(k_dv)`	number of dependent variables	`e(chi2)`	model χ^2
`e(N_g)`	number of groups	`e(p)`	model significance
`e(g_min)`	smallest group size	`e(ic)`	number of iterations
`e(g_avg)`	average group size	`e(rc)`	return code
`e(g_max)`	largest group size		

Macros

`e(cmd)`	`xtnbreg`	`e(method)`	requested estimation method
`e(cmd2)`	`xtn_fe`	`e(user)`	name of likelihood-evaluator program
`e(depvar)`	name of dependent variable	`e(opt)`	type of optimization
`e(title)`	title in estimation output	`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(ivar)`	variable denoting groups	`e(offset)`	offset
`e(wtype)`	weight type	`e(predict)`	program used to implement `predict`
`e(wexp)`	weight expression		

Matrices

`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators

Functions

`e(sample)`	marks estimation sample

`xtnbreg, pa` saves in `e()`:

Scalars

`e(N)`	number of observations	`e(deviance)`	deviance
`e(N_g)`	number of groups	`e(chi2_dev)`	χ^2 test of deviance
`e(g_min)`	smallest group size	`e(dispers)`	deviance dispersion
`e(g_avg)`	average group size	`e(chi2_dis)`	χ^2 test of deviance dispersion
`e(g_max)`	largest group size	`e(tol)`	target tolerance
`e(df_m)`	model degrees of freedom	`e(dif)`	achieved tolerance
`e(chi2)`	model χ^2	`e(phi)`	scale parameter
`e(df_pear)`	degrees of freedom for Pearson χ^2		

Macros

`e(cmd)`	`xtgee`	`e(ivar)`	variable denoting groups
`e(cmd2)`	`xtnbreg`	`e(vcetype)`	covariance estimation method
`e(depvar)`	name of dependent variable	`e(chi2type)`	`Wald`; type of model χ^2 test
`e(family)`	`negative binomial(`k`=1)`	`e(offset)`	offset
`e(link)`	`log`; link function	`e(nbalpha)`	α
`e(corr)`	correlation structure	`e(predict)`	program used to implement `predict`
`e(scale)`	`x2`, `dev`, `phi`, or `#`; scale parameter		

Matrices

`e(b)`	coefficient vector	`e(V)`	variance–covariance matrix of the estimators
`e(R)`	estimated working correlation matrix		

Functions

`e(sample)`	marks estimation sample

Methods and Formulas

`xtnbreg` is implemented as an ado-file.

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

For the random-effects and fixed-effects overdispersion models, we let y_{it} be the count for the t th observation in the ith group. We begin with the model $y_{it} \mid \gamma_{it} \sim \text{Poisson}(\gamma_{it})$, where $\gamma_{it} \mid \delta_i \sim \text{Gamma}(\lambda_{it}, 1/\delta_i)$ with $\lambda_{it} = \exp(\mathbf{x}_{it}\beta + \text{offset}_{it})$ and δ_i is the dispersion parameter. This yields the model

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

Looking at within-group effects only, this specification yields a negative binomial model for the ith group with dispersion (variance divided by the mean) equal to $1 + \delta_i$; i.e., constant dispersion within group. Note that this parameterization of the negative binomial model differs from the default parameterization of `nbreg`, which has dispersion equal to $1 + \alpha\exp(\mathbf{x}\beta + \text{offset})$; 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

$$\begin{aligned}\Pr(Y_{i1} = y_{i1}, \dots, Y_{in_i} = y_{in_i}) &= \int \prod_{t=1}^{n_i} \Pr(Y_{it} = y_{it} \mid \delta_i)\, f(\delta_i)\, d\delta_i \\ &= \frac{\Gamma(r+s)\Gamma(r + \sum_{t=1}^{n_i}\lambda_{it})\Gamma(s + \sum_{t=1}^{n_i} y_{it})}{\Gamma(r)\Gamma(s)\Gamma(r + s + \sum_{t=1}^{n_i}\lambda_{it} + \sum_{t=1}^{n_i} y_{it})} \prod_{t=1}^{n_i} \frac{\Gamma(\lambda_{it}+y_{it})}{\Gamma(\lambda_{it})\Gamma(y_{it}+1)}\end{aligned}$$

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

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 (i.e., the observed $\sum_{t=1}^{n_i} y_{it}$). This yields

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

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 et al. (1984) for a more thorough development of the random-effects and fixed-effects models. Note that Hausman et al. (1984) use a δ that is the inverse of the δ we have used here.

References

Hausman, J., 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.

Liang, K.-Y. and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.

Also See

Complementary: [R] **lincom**, [R] **predict**, [R] **test**, [R] **testnl**, [R] **vce**, [R] **xtdata**, [R] **xtdes**, [R] **xtsum**, [R] **xttab**

Related: [R] **nbreg**, [R] **xtgee**, [R] **xtpois**

Background: [U] **16.5 Accessing coefficients and standard errors**,
[U] **23 Estimation and post-estimation commands**,
[U] **23.11 Obtaining robust variance estimates**,
[R] **xt**