Title

stata.com

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

Description
Quick start
Menu
Syntax
Options for RE model
Options for FE model
Options for PA model
Remarks and examples
Stored results
Methods and formulas
References
Also see

Description

`xtpoisson` fits random-effects, conditional fixed-effects, and population-averaged Poisson models. These models are typically used for a nonnegative count dependent variable.

Quick start

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

```
xtpoisson y x i.a
```

Conditional fixed-effects model with exposure variable `evar`

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

Population-averaged model with robust standard errors

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

As above, but report incidence-rate ratios

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

Menu

Statistics > Longitudinal/panel data > Count outcomes > Poisson regression (FE, RE, PA)

Syntax

Random-effects (RE) model

```
xtpoisson depvar [indepvars] [if] [in] [weight] [, re RE_options]
```

Conditional fixed-effects (FE) model

```
xtpoisson depvar [indepvars] [if] [in] [weight], fe [FE_options]
```

Population-averaged (PA) model

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

RE_options	Description
Model	
noconstant	suppress constant term
re	use random-effects estimator; the default
exposure(*varname*)	include ln(*varname*) in model with coefficient constrained to 1
offset(*varname*)	include *varname* in model with coefficient constrained to 1
normal	use a normal distribution for random effects instead of gamma
constraints(*constraints*)	apply specified linear constraints
collinear	keep collinear variables
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, 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
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite (the default) or ghermite
intpoints(#)	use # quadrature points; default is intpoints(12)
Maximization	
maximize_options	control the maximization process; seldom used
coeflegend	display legend instead of statistics

FE_options	Description
Model	
`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
`collinear`	keep collinear variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`irr`	report incidence-rate ratios
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
maximize_options	control the maximization process; seldom used
`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 xtpoisson, 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, 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 for the random-effects and fixed-effects models; see [U] **11.1.6 weight**. Weights must be constant within panel.

coeflegend does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

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

re, the default, requests the random-effects estimator.

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

normal specifies that the random effects follow a normal distribution instead of a gamma distribution.

constraints(*constraints*), collinear; see [R] **estimation options**.

SE/Robust

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (oim), 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 [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*); see *xtpoisson, re and the robust VCE estimator* in *Methods and formulas*.

Reporting

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

irr reports exponentiated coefficients e^b rather than coefficients b. For the Poisson model, exponentiated coefficients are interpreted as 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**.

Integration

intmethod(*intmethod*), intpoints(#); see [R] **estimation options**. normal must also be specified.

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 option is available with xtpoisson but is not shown in the dialog box:

coeflegend; see [R] **estimation options**.

Options for FE model

Model

`fe` requests the fixed-effects estimator.

`exposure(`*varname*`)`, `offset(`*varname*`)`, `constraints(`*constraints*`)`, `collinear`; see [R] **estimation options**.

SE/Robust

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

`vce(robust)` invokes a cluster–robust estimate of the VCE in which the ID variable specifies the clusters.

Reporting

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

`irr` reports exponentiated coefficients e^b rather than coefficients b. For the Poisson model, exponentiated coefficients are interpreted as incidence-rate ratios.

`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 option is available with `xtpoisson` but is not shown in the dialog box:

`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 Poisson model, exponentiated coefficients are interpreted as 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.

`nolog` suppresses display of the iteration log.

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

The following option is available with `xtpoisson` but is not shown in the dialog box:

`coeflegend`; see [R] **estimation options**.

Remarks and examples

stata.com

`xtpoisson` fits random-effects, conditional fixed-effects, and population-averaged Poisson models. Whenever we refer to a fixed-effects model, we mean the conditional fixed-effects model. These models are typically used for a nonnegative count dependent variable but may be used for any dependent variable in natural logs. For more information about the assumptions of the Poisson model, see [R] **poisson**.

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

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

is equivalent to typing

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

Also see [XT] **xtgee** for information about `xtpoisson`.

By default or when `re` is specified, `xtpoisson` fits via maximum likelihood the random-effects model

$$\Pr(Y_{it} = y_{it}|\mathbf{x}_{it}) = F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

for $i = 1, \ldots, n$ panels, where $t = 1, \ldots, n_i$, and $F(x, z) = \Pr(X = x)$, where X is Poisson distributed with mean $\exp(z)$. In the standard random-effects model, ν_i is assumed to be i.i.d. such that $\exp(\nu_i)$ is gamma with mean one and variance α, which is estimated from the data. If `normal` is specified, ν_i is assumed to be i.i.d. $N(0, \sigma_\nu^2)$.

▷ Example 1

We have data on the number of ship accidents for five different types of ships (McCullagh and Nelder 1989, 205). We wish to analyze whether the "incident" rate is affected by the period in which the ship was constructed and operated. Our measure of exposure is months of service for the ship, and in this model, we assume that the exponentiated random effects are distributed as gamma with mean one and variance α.

```
. use http://www.stata-press.com/data/r15/ships
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) irr
Fitting Poisson model:
Iteration 0:   log likelihood = -147.37993
Iteration 1:   log likelihood = -80.372714
Iteration 2:   log likelihood = -80.116093
Iteration 3:   log likelihood = -80.115916
Iteration 4:   log likelihood = -80.115916
Fitting full model:
Iteration 0:   log likelihood = -79.653186
Iteration 1:   log likelihood = -76.990836  (not concave)
Iteration 2:   log likelihood = -74.824942
Iteration 3:   log likelihood = -74.811243
Iteration 4:   log likelihood = -74.811217
Iteration 5:   log likelihood = -74.811217
Random-effects Poisson regression               Number of obs     =         34
Group variable: ship                            Number of groups  =          5
Random effects u_i ~ Gamma                      Obs per group:
                                                              min =          6
                                                              avg =        6.8
                                                              max =          7
                                                Wald chi2(4)      =      50.90
Log likelihood  = -74.811217                    Prob > chi2       =     0.0000
```

accident	IRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
op_75_79	1.466305	.1734005	3.24	0.001	1.162957	1.848777
co_65_69	2.032543	.304083	4.74	0.000	1.515982	2.72512
co_70_74	2.356853	.3999259	5.05	0.000	1.690033	3.286774
co_75_79	1.641913	.3811398	2.14	0.033	1.04174	2.58786
_cons	.0013724	.0002992	-30.24	0.000	.0008952	.002104
ln(service)	1	(exposure)				
/lnalpha	-2.368406	.8474597			-4.029397	-.7074155
alpha	.0936298	.0793475			.0177851	.4929165

```
Note: Estimates are transformed only in the first equation.
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
LR test of alpha=0: chibar2(01) = 10.61                 Prob >= chibar2 = 0.001
```

The output also includes a likelihood-ratio test of $\alpha = 0$, which compares the panel estimator with the pooled (Poisson) estimator.

We find that the incidence rate for accidents is significantly different for the periods of construction and operation of the ships and that the random-effects model is significantly different from the pooled model.

We may alternatively fit a fixed-effects specification instead of a random-effects specification:

```
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) irr fe

Iteration 0:   log likelihood = -80.738973
Iteration 1:   log likelihood = -54.857546
Iteration 2:   log likelihood = -54.641897
Iteration 3:   log likelihood = -54.641859
Iteration 4:   log likelihood = -54.641859

Conditional fixed-effects Poisson regression    Number of obs     =         34
Group variable: ship                            Number of groups  =          5

                                                Obs per group:
                                                              min =          6
                                                              avg =        6.8
                                                              max =          7

                                                Wald chi2(4)      =      48.44
Log likelihood  = -54.641859                    Prob > chi2       =     0.0000
```

accident	IRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
op_75_79	1.468831	.1737218	3.25	0.001	1.164926	1.852019
co_65_69	2.008002	.3004803	4.66	0.000	1.497577	2.692398
co_70_74	2.26693	.384865	4.82	0.000	1.625274	3.161912
co_75_79	1.573695	.3669393	1.94	0.052	.9964273	2.485397
ln(service)	1	(exposure)				

Both of these models fit the same thing but will differ in efficiency, depending on whether the assumptions of the random-effects model are true.

We could have assumed that the random effects followed a normal distribution, $N(0, \sigma_\nu^2)$, instead of a "log-gamma" distribution, and obtained

```
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) irr
> normal nolog

Random-effects Poisson regression               Number of obs     =         34
Group variable: ship                            Number of groups  =          5

Random effects u_i ~ Gaussian                   Obs per group:
                                                              min =          6
                                                              avg =        6.8
                                                              max =          7

Integration method: mvaghermite                 Integration pts.  =         12

                                                Wald chi2(4)      =      50.95
Log likelihood  = -74.780982                    Prob > chi2       =     0.0000
```

accident	IRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
op_75_79	1.466677	.1734403	3.24	0.001	1.163259	1.849236
co_65_69	2.032604	.3040933	4.74	0.000	1.516025	2.725205
co_70_74	2.357045	.3998397	5.05	0.000	1.690338	3.286717
co_75_79	1.646935	.3820235	2.15	0.031	1.045278	2.594905
_cons	.0013075	.0002775	-31.28	0.000	.0008625	.001982
ln(service)	1	(exposure)				
/lnsig2u	-2.351868	.8586262			-4.034745	-.6689918
sigma_u	.3085306	.1324562			.1330045	.7156988

```
Note: Estimates are transformed only in the first equation.
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
LR test of sigma_u=0: chibar2(01) = 10.67              Prob >= chibar2 = 0.001
```

The output includes the additional panel-level variance component. This is parameterized as the log of the variance $\ln(\sigma_\nu^2)$ (labeled `lnsig2u` in the output). The standard deviation σ_ν is also included in the output labeled `sigma_u`.

When `sigma_u` is zero, the panel-level variance component is unimportant and the panel estimator is no different from the pooled estimator. A likelihood-ratio test of this is included at the bottom of the output. This test formally compares the pooled estimator (`poisson`) with the panel estimator. Here σ_ν is significantly greater than zero, so a panel estimator is indicated. ◁

▷ Example 2

This time we fit a robust equal-correlation population-averaged model:

```
. xtpoisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service) pa
> vce(robust) eform

Iteration 1: tolerance = .04083192
Iteration 2: tolerance = .00270188
Iteration 3: tolerance = .00030663
Iteration 4: tolerance = .00003466
Iteration 5: tolerance = 3.891e-06
Iteration 6: tolerance = 4.359e-07

GEE population-averaged model                   Number of obs     =         34
Group variable:                         ship    Number of groups  =          5
Link:                                    log    Obs per group:
Family:                              Poisson                   min =          6
Correlation:                   exchangeable                    avg =        6.8
                                                               max =          7
                                                Wald chi2(4)      =     252.94
Scale parameter:                            1   Prob > chi2       =     0.0000
                                              (Std. Err. adjusted for clustering on ship)
------------------------------------------------------------------------------
             |                 Robust
    accident |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    op_75_79 |   1.483299   .1197901     4.88   0.000     1.266153    1.737685
    co_65_69 |   2.038477   .1809524     8.02   0.000     1.712955    2.425859
    co_70_74 |   2.643467   .4093947     6.28   0.000     1.951407    3.580962
    co_75_79 |   1.876656     .33075     3.57   0.000     1.328511    2.650966
       _cons |   .0010255   .0000721   -97.90   0.000     .0008935     .001177
ln(service) |          1  (exposure)
------------------------------------------------------------------------------
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
```

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

```
. poisson accident op_75_79 co_65_69 co_70_74 co_75_79, exp(service)
> vce(cluster ship) irr
Iteration 0:   log pseudolikelihood = -147.37993
Iteration 1:   log pseudolikelihood = -80.372714
Iteration 2:   log pseudolikelihood = -80.116093
Iteration 3:   log pseudolikelihood = -80.115916
Iteration 4:   log pseudolikelihood = -80.115916
Poisson regression                              Number of obs     =         34
                                                Wald chi2(3)      =          .
                                                Prob > chi2       =          .
Log pseudolikelihood = -80.115916               Pseudo R2         =     0.3438
                                  (Std. Err. adjusted for 5 clusters in ship)
------------------------------------------------------------------------------
             |               Robust
    accident |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    op_75_79 |    1.47324   .1287036     4.44   0.000       1.2414    1.748377
    co_65_69 |   2.125914   .2850531     5.62   0.000     1.634603    2.764897
    co_70_74 |   2.860138   .6213563     4.84   0.000     1.868384    4.378325
    co_75_79 |   2.021926   .4265285     3.34   0.001     1.337221    3.057227
       _cons |   .0009609   .0000277  -240.66   0.000      .000908    .0010168
ln(service) |          1  (exposure)
------------------------------------------------------------------------------
Note: _cons estimates baseline incidence rate.
```

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtpoisson, re normal` likelihood function is calculated by Gauss–Hermite quadrature, on large problems the computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

`xtpoisson, 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(N_clust)`	number of clusters
`e(alpha)`	value of alpha
`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(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)`	`xtpoisson`
`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(clustvar)`	name of cluster variable
`e(offset)`	linear offset variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(chi2_ct)`	`Wald` or `LR`; type of model χ^2 test corresponding to `e(chi2_c)`
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. Err.
`e(method)`	requested estimation method
`e(distrib)`	`Gamma`; 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
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

xtpoisson, re normal 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(N_clust)	number of clusters
e(sigma_u)	panel-level standard deviation
e(n_quad)	number of quadrature points
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(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)	xtpoisson
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(clustvar)	name of cluster variable
e(offset)	linear offset variable
e(chi2type)	Wald or LR; type of model χ^2 test
e(chi2_ct)	Wald or LR; type of model χ^2 test corresponding to e(chi2_c)
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(intmethod)	integration method
e(distrib)	Gaussian; 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(marginsdefault)	default predict() specification for margins
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
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

xtpoisson, 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_c)	log likelihood, comparison 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)	xtpoisson
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)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
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

xtpoisson, 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)	xtpoisson
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)	Poisson
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)	covariance estimation method
e(nmp)	nmp, if specified
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

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

xtpoisson, fe with robust standard errors implements the formula presented in Wooldridge (1999). The formula is a cluster–robust estimate of the VCE in which the ID variable specifies the clusters.

Although Hausman, Hall, and Griliches (1984) wrote the seminal article on the random-effects and fixed-effects models, Cameron and Trivedi (2013) provide a good textbook treatment. Allison (2009, chap. 4) succinctly discusses these models and illustrates the differences between them using Stata.

For a random-effects specification, we know that

$$\Pr(y_{i1},\ldots,y_{in_i}|\alpha_i,\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i}) = \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_{t=1}^{n_i}\lambda_{it}\right\}\exp\left(\alpha_i\sum_{t=1}^{n_i}y_{it}\right)$$

where $\lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta})$. We may rewrite the above as [defining $\epsilon_i = \exp(\alpha_i)$]

$$\begin{aligned}\Pr(y_{i1},\ldots,y_{in_i}|\epsilon_i,\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i}) &= \left\{\prod_{t=1}^{n_i}\frac{(\lambda_{it}\epsilon_i)^{y_{it}}}{y_{it}!}\right\}\exp\left(-\sum_{t=1}^{n_i}\lambda_{it}\epsilon_i\right)\\ &= \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\exp\left(-\epsilon_i\sum_{t=1}^{n_i}\lambda_{it}\right)\epsilon_i^{\sum_{t=1}^{n_i}y_{it}}\end{aligned}$$

We now assume that ϵ_i follows a gamma distribution with mean one and variance $1/\theta$ so that unconditional on ϵ_i

$$\begin{aligned}\Pr(y_{i1},\ldots,y_{in_i}|\mathbf{X}_i) &= \frac{\theta^\theta}{\Gamma(\theta)}\left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\int_0^\infty \exp\left(-\epsilon_i\sum_{t=1}^{n_i}\lambda_{it}\right)\epsilon_i^{\sum_{t=1}^{n_i}y_{it}}\epsilon_i^{\theta-1}\exp(-\theta\epsilon_i)d\epsilon_i\\ &= \frac{\theta^\theta}{\Gamma(\theta)}\left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\int_0^\infty \exp\left\{-\epsilon_i\left(\theta+\sum_{t=1}^{n_i}\lambda_{it}\right)\right\}\epsilon_i^{\theta+\sum_{t=1}^{n_i}y_{it}-1}d\epsilon_i\\ &= \left(\prod_{t=1}^{n_i}\frac{\lambda_{it}^{y_{it}}}{y_{it}!}\right)\frac{\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)}{\Gamma(\theta)}\left(\frac{\theta}{\theta+\sum_{t=1}^{n_i}\lambda_{it}}\right)^\theta\left(\frac{1}{\theta+\sum_{t=1}^{n_i}\lambda_{it}}\right)^{\sum_{t=1}^{n_i}y_{it}}\end{aligned}$$

for $\mathbf{X}_i = (\mathbf{x}_{i1},\ldots,\mathbf{x}_{in_i})$.

The log likelihood (assuming gamma heterogeneity) is then derived using

$$u_i = \frac{\theta}{\theta+\sum_{t=1}^{n_i}\lambda_{it}} \qquad \lambda_{it} = \exp(\mathbf{x}_{it}\boldsymbol{\beta})$$

$$\Pr(Y_{i1}=y_{i1},\ldots,Y_{in_i}=y_{in_i}|\mathbf{X}_i) = \frac{\prod_{t=1}^{n_i}\lambda_{it}^{y_{it}}\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)}{\prod_{t=1}^{n_i}y_{it}!\Gamma(\theta)\left(\sum_{t=1}^{n_i}\lambda_{it}\right)^{\sum_{t=1}^{n_i}y_{it}}}u_i^\theta(1-u_i)^{\sum_{t=1}^{n_i}y_{it}}$$

such that the log likelihood may be written as

$$\begin{aligned}L = \sum_{i=1}^{n}w_i\Bigg\{&\log\Gamma\left(\theta+\sum_{t=1}^{n_i}y_{it}\right)-\sum_{t=1}^{n_i}\log\Gamma\left(1+y_{it}\right)-\log\Gamma(\theta)+\theta\log u_i\\ &+\log(1-u_i)\sum_{t=1}^{n_i}y_{it}+\sum_{t=1}^{n_i}y_{it}(\mathbf{x}_{it}\boldsymbol{\beta})-\left(\sum_{t=1}^{n_i}y_{it}\right)\log\left(\sum_{t=1}^{n_i}\lambda_{it}\right)\Bigg\}\end{aligned}$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

Alternatively, if we assume a normal distribution, $N(0, \sigma_\nu^2)$, for the random effects ν_i

$$\Pr(y_{i1}, \ldots, y_{in_i}|\mathbf{X}_i) = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

where

$$F(y, z) = \exp\Big\{ -\exp(z) + yz - \log(y!) \Big\}.$$

The panel-level likelihood l_i is given by

$$l_i = \int_{-\infty}^{\infty} \frac{e^{-\nu_i^2/2\sigma_\nu^2}}{\sqrt{2\pi}\sigma_\nu} \left\{ \prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i) \right\} d\nu_i$$

$$\equiv \int_{-\infty}^{\infty} g(y_{it}, x_{it}, \nu_i) d\nu_i$$

This integral can be approximated with M-point Gauss–Hermite quadrature

$$\int_{-\infty}^{\infty} e^{-x^2} h(x) dx \approx \sum_{m=1}^{M} w_m^* h(a_m^*)$$

This is equivalent to

$$\int_{-\infty}^{\infty} f(x) dx \approx \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} f(a_m^*)$$

where the w_m^* denote the quadrature weights and the a_m^* denote the quadrature abscissas. The log likelihood, L, is the sum of the logs of the panel-level likelihoods l_i.

The default approximation of the log likelihood is by adaptive Gauss–Hermite quadrature, which approximates the panel-level likelihood with

$$l_i \approx \sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\left\{(a_m^*)^2\right\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)$$

where $\widehat{\sigma}_i$ and $\widehat{\mu}_i$ are the adaptive parameters for panel i. Therefore, with the definition of $g(y_{it}, x_{it}, \nu_i)$, the total log likelihood is approximated by

$$L \approx \sum_{i=1}^{n} w_i \log \left[\sqrt{2}\widehat{\sigma}_i \sum_{m=1}^{M} w_m^* \exp\big\{(a_m^*)^2\big\} \frac{\exp\big\{-(\sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i)^2/2\sigma_\nu^2\big\}}{\sqrt{2\pi}\sigma_\nu} \right.$$

$$\left. \prod_{t=1}^{n_i} F(y_{it}, x_{it}\boldsymbol{\beta} + \sqrt{2}\widehat{\sigma}_i a_m^* + \widehat{\mu}_i) \right]$$

where w_i is the user-specified weight for panel i; if no weights are specified, $w_i = 1$.

The default method of adaptive Gauss–Hermite quadrature is to calculate the posterior mean and variance and use those parameters for $\widehat{\mu}_i$ and $\widehat{\sigma}_i$ by following the method of Naylor and Smith (1982), further discussed in Skrondal and Rabe-Hesketh (2004). We start with $\widehat{\sigma}_{i,0} = 1$ and $\widehat{\mu}_{i,0} = 0$, and the posterior means and variances are updated in the kth iteration. That is, at the kth iteration of the optimization for l_i, we use

$$l_{i,k} \approx \sum_{m=1}^{M} \sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1})$$

Letting

$$\tau_{i,m,k-1} = \sqrt{2}\widehat{\sigma}_{i,k-1} a_m^* + \widehat{\mu}_{i,k-1}$$

$$\widehat{\mu}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1}) \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}}$$

and

$$\widehat{\sigma}_{i,k} = \sum_{m=1}^{M} (\tau_{i,m,k-1})^2 \frac{\sqrt{2}\widehat{\sigma}_{i,k-1} w_m^* \exp\{(a_m^*)^2\} g(y_{it}, x_{it}, \tau_{i,m,k-1})}{l_{i,k}} - (\widehat{\mu}_{i,k})^2$$

and this is repeated until $\widehat{\mu}_{i,k}$ and $\widehat{\sigma}_{i,k}$ have converged for this iteration of the maximization algorithm. This adaptation is applied on every iteration until the log-likelihood change from the preceding iteration is less than a relative difference of 1e–6; after this, the quadrature parameters are fixed.

The log likelihood can also be calculated by nonadaptive Gauss–Hermite quadrature, the `intmethod(ghermite)` option, where $\rho = \sigma_\nu^2/(\sigma_\nu^2 + 1)$:

$$\begin{aligned} L &= \sum_{i=1}^{n} w_i \log\Big\{\Pr(y_{i1}, \ldots, y_{in_i} | \mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i})\Big\} \\ &\approx \sum_{i=1}^{n} w_i \log\left[\frac{1}{\sqrt{\pi}} \sum_{m=1}^{M} w_m^* \prod_{t=1}^{n_i} F\left\{y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + a_m^* \left(\frac{2\rho}{1-\rho}\right)^{1/2}\right\}\right] \end{aligned}$$

Both quadrature formulas require that the integrated function be well approximated by a polynomial of degree equal to the number of quadrature points. The number of periods (panel size) can affect whether

$$\prod_{t=1}^{n_i} F(y_{it}, \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

is well approximated by a polynomial. As panel size and ρ increase, the quadrature approximation can become less accurate. For large ρ, the random-effects model can also become unidentified. Adaptive quadrature gives better results for correlated data and large panels than nonadaptive quadrature; however, we recommend that you use the `quadchk` command (see [XT] **quadchk**) to verify the quadrature approximation used in this command, whichever approximation you choose.

For a fixed-effects specification, we know that

$$\begin{aligned}\Pr(Y_{it}=y_{it}|\mathbf{x}_{it}) &= \exp\{-\exp(\alpha_i+\mathbf{x}_{it}\boldsymbol{\beta})\}\exp(\alpha_i+\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}/y_{it}!\\ &= \frac{1}{y_{it}!}\exp\{-\exp(\alpha_i)\exp(\mathbf{x}_{it}\boldsymbol{\beta})+\alpha_i y_{it}\}\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}\\ &\equiv F_{it}\end{aligned}$$

Because we know that the observations are independent, we may write the joint probability for the observations within a panel as

$$\begin{aligned}&\Pr\left(Y_{i1}=y_{i1},\ldots,Y_{in_i}=y_{in_i}|\mathbf{X}_i\right)\\ &\qquad=\prod_{t=1}^{n_i}\frac{1}{y_{it}!}\exp\{-\exp(\alpha_i)\exp(\mathbf{x}_{it}\boldsymbol{\beta})+\alpha_i y_{it}\}\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}\\ &\qquad=\left(\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta})+\alpha_i\sum_t y_{it}\right\}\end{aligned}$$

and we also know that the sum of n_i Poisson independent random variables, each with parameter λ_{it} for $t=1,\ldots,n_i$, is distributed as Poisson with parameter $\sum_t\lambda_{it}$. Thus

$$\Pr\left(\sum_t Y_{it}=\sum_t y_{it}\Big|\mathbf{X}_i\right)=$$

$$\frac{1}{(\sum_t y_{it})!}\exp\left\{-\exp(\alpha_i)\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta})+\alpha_i\sum_t y_{it}\right\}\left\{\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta})\right\}^{\sum_t y_{it}}$$

So, the conditional likelihood is conditioned on the sum of the outcomes in the set (panel). The appropriate function is given by

$$\begin{aligned}&\Pr\left(Y_{i1}=y_{i1},\ldots,Y_{in_i}=y_{in_i}\Big|\mathbf{X}_i,\sum_t Y_{it}=\sum_t y_{it}\right)=\\ &\qquad\left[\left(\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!}\right)\exp\left\{-\exp(\alpha_i)\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta})+\alpha_i\sum_t y_{it}\right\}\right]\Big/\\ &\qquad\left[\frac{1}{(\sum_t y_{it})!}\exp\left\{-\exp(\alpha_i)\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta})+\alpha_i\sum_t y_{it}\right\}\left\{\sum_t\exp(\mathbf{x}_{it}\boldsymbol{\beta})\right\}^{\sum_t y_{it}}\right]\\ &\qquad=\left(\sum_t y_{it}\right)!\prod_{t=1}^{n_i}\frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}!\left\{\sum_k\exp(\mathbf{x}_{ik}\boldsymbol{\beta})\right\}^{y_{it}}}\end{aligned}$$

which is free of α_i.

The conditional log likelihood is given by

$$
\begin{aligned}
L &= \log \prod_{i=1}^{n} \left[\left(\sum_{t=1}^{n_i} y_{it} \right)! \prod_{t=1}^{n_i} \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta})^{y_{it}}}{y_{it}! \left\{ \sum_{\ell=1}^{n_\ell} \exp(\mathbf{x}_{i\ell}\boldsymbol{\beta}) \right\}^{y_{it}}} \right]^{w_i} \\
&= \log \prod_{i=1}^{n} \left\{ \frac{(\sum_t y_{it})!}{\prod_{t=1}^{n_i} y_{it}!} \prod_{t=1}^{n_i} p_{it}^{y_{it}} \right\}^{w_i} \\
&= \sum_{i=1}^{n} w_i \left\{ \log\Gamma \left(\sum_{t=1}^{n_i} y_{it} + 1 \right) - \sum_{t=1}^{n_i} \log\Gamma(y_{it}+1) + \sum_{t=1}^{n_i} y_{it} \log p_{it} \right\}
\end{aligned}
$$

where

$$
p_{it} = e^{\mathbf{x}_{it}\boldsymbol{\beta}} \Big/ \sum_{\ell} e^{\mathbf{x}_{i\ell}\boldsymbol{\beta}}
$$

xtpoisson, re and the robust VCE estimator

Specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the coefficients estimated in this regression. See [P] **_robust**, particularly *Introduction* and *Methods and formulas*. Wooldridge (2016) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2016), Stock and Watson (2008), and Arellano (2003), specifying `vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`, where *panelvar* is the variable that identifies the panels.

Clustering on the panel variable produces a consistent VCE estimator when the disturbances are not identically distributed over the panels or there is serial correlation in ϵ_{it}.

The cluster–robust VCE estimator requires that there are many clusters and the disturbances are uncorrelated across the clusters. The panel variable must be nested within the cluster variable because of the within-panel correlation that is generally induced by the random-effects transform when there is heteroskedasticity or within-panel serial correlation in the idiosyncratic errors.

References

Allison, P. D. 2009. *Fixed Effects Regression Models*. Newbury Park, CA: Sage.

Arellano, M. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press.

Baltagi, B. H. 2009. *A Companion to Econometric Analysis of Panel Data*. Chichester, UK: Wiley.

——. 2013. *Econometric Analysis of Panel Data*. 5th ed. Chichester, UK: Wiley.

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

Cummings, P. 2011. Estimating adjusted risk ratios for matched and unmatched data: An update. *Stata Journal* 11: 290–298.

Greene, W. H. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Hardin, J. W., and J. M. Hilbe. 2012. *Generalized Linear Models and Extensions*. 3rd ed. College Station, TX: Stata Press.

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.

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

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society, Series C* 31: 214–225.

Skrondal, A., and S. Rabe-Hesketh. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Boca Raton, FL: Chapman & Hall/CRC.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174.

Wooldridge, J. M. 1999. Distribution-free estimation of some nonlinear panel data models. *Journal of Econometrics* 90: 77–97.

——. 2016. *Introductory Econometrics: A Modern Approach*. 6th ed. Boston: Cengage.

Also see

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

[XT] **quadchk** — Check sensitivity of quadrature approximation

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

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

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

[ME] **mepoisson** — Multilevel mixed-effects Poisson regression

[ME] **meqrpoisson** — Multilevel mixed-effects Poisson regression (QR decomposition)

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

[R] **poisson** — Poisson regression

[U] 20 Estimation and postestimation commands