

**xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

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

## Description

`xtlogit` fits random-effects, conditional fixed-effects, and population-averaged logit models for a binary dependent variable. The probability of a positive outcome is assumed to be determined by the logistic cumulative distribution function. Results may be reported as coefficients or odds ratios.

## Quick start

Random-effects model of  $y$  as a function of  $x_1$ ,  $x_2$ , and [indicators](#) for levels of categorical variable  $a$  using `xtset` data

```
xtlogit y x1 x2 i.a
```

As above, but report odds ratios

```
xtlogit y x1 x2 i.a, or
```

Conditional fixed-effects model

```
xtlogit y x1 x2 i.a, fe
```

Population-averaged model with robust standard errors

```
xtlogit y x1 x2 i.a, pa vce(robust)
```

Random-effects model with cluster-robust standard errors for panels nested within `cvar`

```
xtlogit y x1 x2 i.a, vce(cluster cvar)
```

## Menu

Statistics > Longitudinal/panel data > Binary outcomes > Logistic regression (FE, RE, PA)

## Syntax

*Random-effects (RE) model*

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

*Conditional fixed-effects (FE) model*

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

*Population-averaged (PA) model*

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

*RE\_options*

Description

Model

`noconstant`

suppress constant term

`re`

use random-effects estimator; the default

`offset(varname)`

include *varname* in model with coefficient constrained to 1

`constraints(constraints)`

apply specified linear constraints

`collinear`

keep collinear variables

`asis`

retain perfect predictor variables

SE/Robust

`vce(vcetype)`

*vcetype* may be `oim`, `robust`, `cluster clustvar`, `bootstrap`, or `jackknife`

Reporting

`level(#)`

set confidence level; default is `level(95)`

`or`

report odds 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

`nodisplay`

suppress display of header and coefficients

`coeflegend`

display legend instead of statistics

<i>FE_options</i>	Description
Model	
<code>fe</code>	use fixed-effects estimator
<code>offset(<i>varname</i>)</code>	include <i>varname</i> in model with coefficient constrained to 1
<code>constraints(<i>constraints</i>)</code>	apply specified linear constraints
<code>collinear</code>	keep collinear variables
SE	
<code>vce(<i>vcetype</i>)</code>	<i>vcetype</i> may be <code>oim</code> , <code>bootstrap</code> , or <code>jackknife</code>
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>or</code>	report odds ratios
<code>lrmodel</code>	perform the likelihood-ratio model test instead of the default Wald test
<code>nocnsreport</code>	do not display constraints
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>nodisplay</code>	suppress display of header and coefficients
<code>coeflegend</code>	display legend instead of statistics

---

## 4 xtlogit — Fixed-effects, random-effects, and population-averaged logit models

---

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

---

<i>correlation</i>	Description
<u>exchangeable</u>	exchangeable
<u>independent</u>	independent
<u>unstructured</u>	unstructured
<u>fixed matname</u>	user-specified
<u>ar #</u>	autoregressive of order #
<u>stationary #</u>	stationary of order #
<u>nonstationary #</u>	nonstationary of order #

A panel variable must be specified. For `xtlogit`, `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 fixed-effects and random-effects models; see [U] [11.1.6 weight](#). Weights must be constant within panel.

`nodisplay` 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 model

### Model

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

`re` requests the random-effects estimator, which is the default.

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

`asis` forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] [probit](#).

### SE/Robust

`vce(vctype)` 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 [xtlogit](#), [re](#) and [the robust VCE estimator](#) in *Methods and formulas*.

### Reporting

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

`or` reports the estimated coefficients transformed to odds ratios, that is,  $e^b$  rather than  $b$ . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `or` may be specified at estimation or when replaying previously estimated results.

`lmodel`, `nocnsreport`; see [R] [estimation options](#).

*display\_options*: `nocl`, `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](#).

### 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 `xtlogit` but are not shown in the dialog box:

`nodisplay` is for programmers. It suppresses the display of the header and the coefficients.  
`coeflegend`; see [R] [estimation options](#).

## Options for FE model

### Model

---

`fe` requests the fixed-effects estimator.

`offset(varname)`, `constraints(constraints)`, `collinear`; 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](#).

`or` reports the estimated coefficients transformed to odds ratios, that is,  $e^b$  rather than  $b$ . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `or` may be specified at estimation or when replaying previously estimated results.

`lrmodel`, `nocnsreport`; see [R] [estimation options](#).

*display\_options*: `nocl`, `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(#)`, `nonrntolerance`, and `from(init_specs)`; see [R] [maximize](#). These options are seldom used.

The following options are available with `xtlogit` but are not shown in the dialog box:

`nodisplay` is for programmers. It suppresses the display of the header and the coefficients.

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

## Options for PA model

## Model

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

`pa` requests the population-averaged estimator.

`offset(varname)`; see [R] [estimation options](#).

`asis` forces retention of perfect predictor variables and their associated, perfectly predicted observations and may produce instabilities in maximization; see [R] [probit](#).

## 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](#).

or reports the estimated coefficients transformed to odds ratios, that is,  $e^b$  rather than  $b$ . Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `or` may be specified at estimation or when replaying previously estimated results.

*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 options are available with `xtlogit` but are not shown in the dialog box:

`nodisplay` is for programmers. It suppresses the display of the header and the coefficients.

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

## Remarks and examples

[stata.com](http://www.stata.com)

`xtlogit` fits random-effects, conditional fixed-effects, and population-averaged logit models. Whenever we refer to a fixed-effects model, we mean the conditional fixed-effects model. *depvar* equal to nonzero and nonmissing (typically *depvar* equal to one) indicates a positive outcome, whereas *depvar* equal to zero indicates a negative outcome.

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

```
. xtlogit ..., pa ...
```

is equivalent to typing

```
. xtgee ..., ... family(binomial) link(logit) corr(exchangeable)
```

It is also a convenience command if you want the fixed-effects model. Typing

```
. xtlogit ..., fe ...
```

is equivalent to typing

```
. clogit ..., group(varname_i) ...
```

See also [XT] [xtgee](#) and [R] [clogit](#) for information about `xtlogit`.



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

$$\Pr(y_{it} \neq 0 | \mathbf{x}_{it}) = P(\mathbf{x}_{it}\boldsymbol{\beta} + \nu_i)$$

for  $i = 1, \dots, n$  panels, where  $t = 1, \dots, n_i$ ,  $\nu_i$  are i.i.d.,  $N(0, \sigma_\nu^2)$ , and  $P(z) = \{1 + \exp(-z)\}^{-1}$ .

Underlying this model is the variance components model

$$y_{it} \neq 0 \iff \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} > 0$$

where  $\epsilon_{it}$  are i.i.d. logistic distributed with mean zero and variance  $\sigma_\epsilon^2 = \pi^2/3$ , independently of  $\nu_i$ .

### ► Example 1

We are studying unionization of women in the United States and are using the `union` dataset; see [\[XT\] xt](#). We wish to fit a random-effects model of union membership:

```
. use http://www.stata-press.com/data/r15/union
(NLS Women 14-24 in 1968)
. xtlogit union age grade not_smsa south##c.year
(output omitted)
```

```
Random-effects logistic regression      Number of obs      =      26,200
Group variable: idcode                 Number of groups   =       4,434
Random effects u_i ~ Gaussian          Obs per group:
                                         min =              1
                                         avg =              5.9
                                         max =              12

Integration method: mvaghermite        Integration pts.   =       12
                                         Wald chi2(6)      =      227.46
Log likelihood = -10540.274             Prob > chi2       =       0.0000
```

union	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
age	.0156732	.0149895	1.05	0.296	-.0137056	.045052
grade	.0870851	.0176476	4.93	0.000	.0524965	.1216738
not_smsa	-.2511884	.0823508	-3.05	0.002	-.4125929	-.0897839
1.south	-2.839112	.6413116	-4.43	0.000	-4.096059	-1.582164
year	-.0068604	.0156575	-0.44	0.661	-.0375486	.0238277
south#c.year						
1	.0238506	.0079732	2.99	0.003	.0082235	.0394777
_cons	-3.009365	.8414963	-3.58	0.000	-4.658667	-1.360062
/lnsig2u	1.749366	.0470017			1.657245	1.841488
sigma_u	2.398116	.0563577			2.290162	2.511158
rho	.6361098	.0108797			.6145307	.6571548

LR test of rho=0: `chibar2(01) = 6004.43` Prob >= `chibar2 = 0.000`

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  $\sigma_\nu$  is also included in the output and labeled `sigma_u` together with  $\rho$  (labeled `rho`),

$$\rho = \frac{\sigma_\nu^2}{\sigma_\nu^2 + \sigma_\epsilon^2}$$

which is the proportion of the total variance contributed by the panel-level variance component.

When rho 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 (logit) with the panel estimator.

As an alternative to the random-effects specification, we might want to fit an equal-correlation logit model:

```
. xtlogit union age grade not_smsa south##c.year, pa
Iteration 1: tolerance = .1487877
Iteration 2: tolerance = .00949342
Iteration 3: tolerance = .00040606
Iteration 4: tolerance = .00001602
Iteration 5: tolerance = 6.628e-07

GEE population-averaged model
Group variable:          idcode      Number of obs   =    26,200
Link:                   logit        Number of groups =     4,434
Family:                 binomial     Obs per group:
Correlation:            exchangeable   min =           1
                                                avg =           5.9
                                                max =           12
                                                Wald chi2(6)    =    235.08
Scale parameter:        1            Prob > chi2      =     0.0000
```

union	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
age	.0165893	.0092229	1.80	0.072	-.0014873	.0346659
grade	.0600669	.0108343	5.54	0.000	.0388321	.0813016
not_smsa	-.1215445	.0483713	-2.51	0.012	-.2163505	-.0267384
1.south	-1.857094	.372967	-4.98	0.000	-2.588096	-1.126092
year	-.0121168	.0095707	-1.27	0.205	-.030875	.0066413
south#c.year						
1	.0160193	.0046076	3.48	0.001	.0069886	.0250501
_cons	-1.39755	.5089508	-2.75	0.006	-2.395075	-.4000247

▷ Example 2

xtlogit with the pa option allows a vce(robust) option, so we can obtain the population-averaged logit estimator with the robust variance calculation by typing

```
. xtlogit union age grade not_smsa south#c.year, pa vce(robust) nolog
GEE population-averaged model          Number of obs   =   26,200
Group variable:                        idcode          Number of groups =    4,434
Link:                                   logit            Obs per group:
Family:                                 binomial                min =         1
Correlation:                            exchangeable          avg =         5.9
                                                max =         12
                                                Wald chi2(6)         =   154.88
Scale parameter:                        1                Prob > chi2          =    0.0000
                                                (Std. Err. adjusted for clustering on idcode)
```

union	Coef.	Robust Std. Err.	z	P> z	[95% Conf. Interval]	
age	.0165893	.008951	1.85	0.064	-.0009543	.0341329
grade	.0600669	.0133193	4.51	0.000	.0339616	.0861722
not_smsa	-.1215445	.0613803	-1.98	0.048	-.2418477	-.0012412
1.south	-1.857094	.5389238	-3.45	0.001	-2.913366	-.8008231
year	-.0121168	.0096998	-1.25	0.212	-.0311282	.0068945
south#c.year						
1	.0160193	.0067217	2.38	0.017	.002845	.0291937
_cons	-1.39755	.5603767	-2.49	0.013	-2.495868	-.2992317

These standard errors are somewhat larger than those obtained without the vce(robust) option.

Finally, we can also fit a fixed-effects model to these data (see also [\[R\] clogit](#) for details):

```
. xtlogit union age grade not_smsa south##c.year, fe
note: multiple positive outcomes within groups encountered.
note: 2,744 groups (14,165 obs) dropped because of all positive or
      all negative outcomes.

Iteration 0:  log likelihood = -4516.5881
Iteration 1:  log likelihood = -4510.8906
Iteration 2:  log likelihood = -4510.888
Iteration 3:  log likelihood = -4510.888

Conditional fixed-effects logistic regression
Group variable: idcode

Number of obs      =      12,035
Number of groups   =      1,690

Obs per group:
      min =          2
      avg =          7.1
      max =          12

LR chi2(6)        =      78.60
Prob > chi2       =      0.0000

Log likelihood    = -4510.888
```

union	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
age	.0710973	.0960536	0.74	0.459	-.1171643	.2593589
grade	.0816111	.0419074	1.95	0.051	-.0005259	.163748
not_smsa	.0224809	.1131786	0.20	0.843	-.199345	.2443069
1.south	-2.856488	.6765694	-4.22	0.000	-4.182539	-1.530436
year	-.0636853	.0967747	-0.66	0.510	-.2533602	.1259896
south#c.year						
1	.0264136	.0083216	3.17	0.002	.0101036	.0427235

◀

## □ 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 `xtlogit` 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

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

### Scalars

<code>e(N)</code>	number of observations
<code>e(N_g)</code>	number of groups
<code>e(k)</code>	number of parameters
<code>e(k_aux)</code>	number of auxiliary parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model
<code>e(ll_c)</code>	log likelihood, comparison model
<code>e(chi2)</code>	$\chi^2$
<code>e(chi2_c)</code>	$\chi^2$ for comparison test
<code>e(N_clust)</code>	number of clusters
<code>e(rho)</code>	$\rho$
<code>e(sigma_u)</code>	panel-level standard deviation
<code>e(n_quad)</code>	number of quadrature points
<code>e(g_min)</code>	smallest group size
<code>e(g_avg)</code>	average group size
<code>e(g_max)</code>	largest group size
<code>e(p)</code>	$p$ -value for model test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(rank0)</code>	rank of <code>e(V)</code> for constant-only model
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	xtlogit
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(ivar)</code>	variable denoting groups
<code>e(model)</code>	re
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(clustvar)</code>	name of cluster variable
<code>e(offset)</code>	linear offset variable
<code>e(chi2type)</code>	Wald or LR; type of model $\chi^2$ test
<code>e(chi2_ct)</code>	Wald or LR; type of model $\chi^2$ test corresponding to <code>e(chi2_c)</code>
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. Err.
<code>e(intmethod)</code>	integration method
<code>e(distrib)</code>	Gaussian; the distribution of the random effect
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsdefault)</code>	default <code>predict()</code> specification for <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

## Matrices

<code>e(b)</code>	coefficient vector
<code>e(Cns)</code>	constraints matrix
<code>e(ilog)</code>	iteration log
<code>e(gradient)</code>	gradient vector
<code>e(V)</code>	variance–covariance matrix of the estimators
<code>e(V_modelbased)</code>	model-based variance

## Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

`xtlogit`, `fe` stores the following in `e()`:

## Scalars

<code>e(N)</code>	number of observations
<code>e(N_g)</code>	number of groups
<code>e(N_drop)</code>	number of observations dropped because of all positive or all negative outcomes
<code>e(N_group_drop)</code>	number of groups dropped because of all positive or all negative outcomes
<code>e(k)</code>	number of parameters
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(r2_p)</code>	pseudo <i>R</i> -squared
<code>e(ll)</code>	log likelihood
<code>e(ll_0)</code>	log likelihood, constant-only model
<code>e(chi2)</code>	$\chi^2$
<code>e(g_min)</code>	smallest group size
<code>e(g_avg)</code>	average group size
<code>e(g_max)</code>	largest group size
<code>e(p)</code>	<i>p</i> -value for model test
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

## Macros

<code>e(cmd)</code>	<code>clogit</code>
<code>e(cmd2)</code>	<code>xtlogit</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(ivar)</code>	variable denoting groups
<code>e(model)</code>	<code>fe</code>
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(title)</code>	title in estimation output
<code>e(offset)</code>	linear offset variable
<code>e(chi2type)</code>	LR; type of model $\chi^2$ test
<code>e(vce)</code>	<i>vctype</i> specified in <code>vce()</code>
<code>e(group)</code>	name of <code>group()</code> variable
<code>e(multiple)</code>	multiple if multiple positive outcomes within groups
<code>e(opt)</code>	type of optimization
<code>e(which)</code>	max or min; whether optimizer is to perform maximization or minimization
<code>e(ml_method)</code>	type of ml method
<code>e(user)</code>	name of likelihood-evaluator program
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(marginsdefault)</code>	default <code>predict()</code> specification for <code>margins</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

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

xtlogit, pa stores the following in e():

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

Macros	
e(cmd)	xtgee
e(cmd2)	xtlogit
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)	binomial
e(link)	logit; 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 $\chi^2$ test
e(vce)	<i>vctype</i> specified in vce()
e(vcetype)	title used to label Std. Err.
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

`xtlogit` reports the population-averaged results obtained using `xtgee`, `family(binomial)` `link(logit)` to obtain estimates. The fixed-effects results are obtained using `clogit`. See [XT] `xtgee` and [R] `clogit` for details on the methods and formulas.

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

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

where

$$F(y, z) = \begin{cases} \frac{1}{1 + \exp(-z)} & \text{if } y \neq 0 \\ \frac{1}{1 + \exp(z)} & \text{otherwise} \end{cases}$$

The panel-level likelihood  $l_i$  is given by

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

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

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

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

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



where  $\hat{\sigma}_i$  and  $\hat{\mu}_i$  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

$$L \approx \sum_{i=1}^n w_i \log \left[ \sqrt{2\hat{\sigma}_i} \sum_{m=1}^M w_m^* \exp\{(a_m^*)^2\} \frac{\exp\{-(\sqrt{2\hat{\sigma}_i}a_m^* + \hat{\mu}_i)^2/2\sigma_\nu^2\}}{\sqrt{2\pi}\sigma_\nu} \right. \\ \left. \prod_{t=1}^{n_i} F(y_{it}, x_{it}\beta + \sqrt{2\hat{\sigma}_i}a_m^* + \hat{\mu}_i) \right]$$

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  $\hat{\mu}_i$  and  $\hat{\sigma}_i$  by following the method of [Naylor and Smith \(1982\)](#), further discussed in [Skrondal and Rabe-Hesketh \(2004\)](#). We start with  $\hat{\sigma}_{i,0} = 1$  and  $\hat{\mu}_{i,0} = 0$ , and the posterior means and variances are updated in the  $k$ th iteration. That is, at the  $k$ th iteration of the optimization for  $l_i$ , we use

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

Letting

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

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

and

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

and this is repeated until  $\hat{\mu}_{i,k}$  and  $\hat{\sigma}_{i,k}$  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 `int-method(ghermite)` option, where  $\rho = \sigma_\nu^2/(\sigma_\nu^2 + 1)$ :

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

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

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

is well approximated by a polynomial. As panel size and  $\rho$  increase, the quadrature approximation can become less accurate. For large  $\rho$ , 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.

## **xlogit, 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  $\epsilon_{it}$ .

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.
- Conway, M. R. 1990. A random effects model for binary data. *Biometrics* 46: 317–328.
- Cruz-Gonzalez, M., I. Fernández-Val, and M. Weidner. 2017. Bias corrections for probit and logit models with two-way fixed effects. *Stata Journal* 17: 517–545.
- Liang, K.-Y., and S. L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73: 13–22.
- 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.
- Neuhaus, J. M. 1992. Statistical methods for longitudinal and clustered designs with binary responses. *Statistical Methods in Medical Research* 1: 249–273.
- Neuhaus, J. M., J. D. Kalbfleisch, and W. W. Hauck. 1991. A comparison of cluster-specific and population-averaged approaches for analyzing correlated binary data. *International Statistical Review* 59: 25–35.
- Pendergast, J. F., S. J. Gange, M. A. Newton, M. J. Lindstrom, M. Palta, and M. R. Fisher. 1996. A survey of methods for analyzing clustered binary response data. *International Statistical Review* 64: 89–118.
- Skrdal, 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.
- Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.
- Wooldridge, J. M. 2016. *Introductory Econometrics: A Modern Approach*. 6th ed. Boston: Cengage.

## Also see

- [XT] **xtlogit postestimation** — Postestimation tools for xtlogit
- [XT] **quadchk** — Check sensitivity of quadrature approximation
- [XT] **xtcloglog** — Random-effects and population-averaged cloglog models
- [XT] **xtgee** — Fit population-averaged panel-data models by using GEE
- [XT] **xtprobit** — Random-effects and population-averaged probit models
- [XT] **xtset** — Declare data to be panel data
- [ME] **melogit** — Multilevel mixed-effects logistic regression
- [ME] **meqrlogit** — Multilevel mixed-effects logistic regression (QR decomposition)
- [MI] **estimation** — Estimation commands for use with mi estimate
- [R] **cllogit** — Conditional (fixed-effects) logistic regression
- [R] **logistic** — Logistic regression, reporting odds ratios
- [R] **logit** — Logistic regression, reporting coefficients
- [U] **20 Estimation and postestimation commands**