Title

xtreg — Fixed-, between-, and random-effects and population-averaged linear models+

+This command includes features that are part of StataNow.

Description

`xtreg` fits regression models to panel data. In particular, `xtreg` with the `be` option fits random-effects models by using the between regression estimator; with the `fe` option, it fits fixed-effects models (by using the within regression estimator); and with the `re` option, it fits random-effects models by using the GLS estimator (producing a matrix-weighted average of the between and within results). See [XT] **xtdata** for a faster way to fit fixed- and random-effects models.

Quick start

Random-effects linear regression by GLS of `y` on `x1` and `x2` using `xtset` data

```
xtreg y x1 x2
```

Same as above, but estimate by maximum likelihood

```
xtreg y x1 x2, mle
```

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

```
xtreg y x1 x2, fe vce(cluster cvar)
```

Fixed-effects model absorbing indicator variables for the levels of `cvar` (StataNow)

```
xtreg y x1 x2, fe absorb(cvar)
```

Population-averaged model with an exchangeable within-panel correlation structure

```
xtreg y x1 x2, pa
```

Same as above, but specify an autoregressive correlation structure of order 1

```
xtreg y x1 x2, pa corr(ar 1)
```

Between-effects model

```
xtreg y x1 x2, be
```

Menu

Statistics > Longitudinal/panel data > Linear models > Linear regression (FE, RE, PA, BE)

Syntax

GLS random-effects (RE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] [, re *RE_options*]

Between-effects (BE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] , be [*BE_options*]

Fixed-effects (FE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , fe [*FE_options*]

ML random-effects (MLE) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , mle [*MLE_options*]

Population-averaged (PA) model

xtreg *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , pa [*PA_options*]

RE_options	Description
Model	
re	use random-effects estimator; the default
sa	use Swamy–Arora estimator of the variance components
SE/Robust	
vce(*vcetype*)	*vcetype* may be conventional, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
theta	report θ
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
coeflegend	display legend instead of statistics

BE_options	Description
Model	
`be`	use between-effects estimator
`wls`	use weighted least squares
SE	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `bootstrap`, or `jackknife`
Reporting	
`level(`*#*`)`	set confidence level; default is `level(95)`
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
`coeflegend`	display legend instead of statistics

FE_options	Description
Model	
`fe`	use fixed-effects estimator
+`absorb(`*varlist* [, *method*]`)`	specify categorical variables to be absorbed
+`nosigmau`	do not estimate σ_u
+`nouitest`	suppress the F test for u_i and do not estimate σ_u
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `conventional`, `robust`, `cluster` *clustvarlist*, `bootstrap`, `jackknife`, or `hc2` [*clustvar*]
Reporting	
`level(`*#*`)`	set confidence level; default is `level(95)`
`clustertable`	display table of multiway cluster combinations
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	
+`iterate(`*#*`)`	maximum number of iterations for alternating projection method (APM); default is `iterate(50)`
+`nolog`	suppress the APM iteration log
+`tolerance(`*#*`)`	convergence tolerance for maximum absolute difference; default is `tolerance(1e-8)`
`coeflegend`	display legend instead of statistics

MLE_options	Description
Model	
noconstant	suppress constant term
mle	use ML random-effects estimator
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
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
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
rgf	multiply the robust variance estimate by $(N-1)/(N-P)$
scale(*parm*)	overrides the default scale parameter; *parm* may be x2, dev, phi, or #
Reporting	
level(#)	set confidence level; default is level(95)
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 #

+These features are part of StataNow.

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

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

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

by, collect, mi estimate, and statsby are allowed; see [U] **11.1.10 Prefix commands**. bayes is allowed for the GLS random-effects model. For more details, see [BAYES] **bayes: xtreg**. fp is allowed for the between-effects, fixed-effects, and maximum-likelihood random-effects models.

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

vce() is not allowed with the by prefix.

aweights, fweights, and pweights are allowed for the fixed-effects model. iweights, fweights, and pweights are allowed for the population-averaged model. iweights are allowed for the maximum-likelihood random-effects (MLE) model. 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

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

sa specifies that the small-sample Swamy–Arora estimator individual-level variance component be used instead of the default consistent estimator. See *xtreg, re* in *Methods and formulas* for details.

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), that allow for intragroup correlation (cluster *clustvar*), 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.

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

Reporting

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

theta specifies that the output include the estimated value of θ used in combining the between and fixed estimators. For balanced data, this is a constant, and for unbalanced data, a summary of the values is presented in the header of the output.

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**.

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

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

Options for BE model

Model

be requests the between regression estimator.

wls specifies that, for unbalanced data, weighted least squares be used rather than the default OLS. Both methods produce consistent estimates. The true variance of the between-effects residual is $\sigma_\nu^2 + T_i\sigma_\epsilon^2$ (see xtreg, be in *Methods and formulas* below). WLS produces a "stabilized" variance of $\sigma_\nu^2/T_i + \sigma_\epsilon^2$, which is also not constant. Thus the choice between OLS and WLS amounts to which is more stable.

Comment: xtreg, be is rarely used anyway, but between estimates are an ingredient in the random-effects estimate. Our implementation of xtreg, re uses the OLS estimates for this ingredient, based on our judgment that σ_ν^2 is large relative to σ_ϵ^2 in most models. Formally, only a consistent estimate of the between estimates is required.

SE

vce(*vcetype*) specifies the type of standard error reported, which includes types that are derived from asymptotic theory (conventional) 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.

Reporting

level(#); 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**.

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

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

Options for FE model

Model

`fe` requests the fixed-effects (within) regression estimator.

`absorb(`*varlist* [`,` *method*]`)` is part of StataNow. It specifies the categorical variables to be absorbed. The results are adjusted as if indicator variables for each level of each variable in *varlist* were included in the regression.

An APM iterative algorithm is used to project the *depvar* and *indepvars* to absorb these variables and the panel variable. *method* specifies the APM and is one of `halperin` or `cimmino`.

`halperin`, the default, uses the product of the projection matrices.

`cimmino` uses the mean of the projection matrices.

The two methods typically perform similarly. See Stammann (2018) for details.

`absorb()` may not be specified with option `vce(hc2)`.

`nosigmau` is part of StataNow. It prevents the computation of σ_u. When `absorb(`*varlist*`)` is specified, the estimation of σ_u requires an iterative search. If σ_u is not of interest, you may specify `nosigmau` to save computational time.

`nouitest` is part of StataNow. It prevents computation of σ_u and the F test that all u_i are jointly zero. If σ_u and the F test are of no interest, you may specify `nouitest` to save computational time.

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`), that allow for intragroup correlations (`cluster` *clustvarlist*), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [XT] ***vce_options***.

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

`vce(robust)` is equivalent to specifying `vce(cluster` *panelvar*`)`.

`vce(cluster` *clustvarlist*`)` specifies that standard errors allow for intragroup correlation within groups defined by one or more variables in *clustvarlist*, relaxing the usual requirement that the observations be independent. For example, `vce(cluster clustvar1)` produces cluster–robust standard errors that allow for observations that are independent across groups defined by `clustvar1` but not necessarily independent within groups. You could also type `vce(cluster clustvar1 clustvar2` ... `cluster`*p*`)` to account for correlation within groups formed by p variables (multiway clustering).

`xtreg` also allows the following:

`vce(hc2` [*clustvar*] [`, dfadjust`]`)` specifies a cluster–robust variance calculation that uses a bias correction that is an alternative to the one provided by `vce(robust)` and `vce(cluster` *clustvarlist*`)`. Specifying `vce(hc2` [`, dfadjust`]`)` is equivalent to specifying `vce(hc2` *panelvar* [`, dfadjust`]`)`. `dfadjust` computes the Bell and McCaffrey (2002) adjusted degrees of freedom based on the clusters of *clustvar*; see *xtreg, fe* in *Methods and formulas*. Note that `dfadjust` does not affect multiple-imputation results when the command is used with `mi estimate`.

`vce(hc2)` may not be specified with option `absorb()`.

Reporting

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

clustertable displays a table reporting cluster combinations and the number of clusters per combination. This option is available only when vce(cluster *clustvarlist)* is specified with more than one variable in *clustvarlist* to compute multiway cluster–robust standard errors.

display_options: noci, nopvalues, dfci, dfpvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(*style*), cformat(%*fmt*), pformat(%*fmt*), sformat(%*fmt*), and nolstretch; see [R] **Estimation options**.

dfci specifies that parameter degrees of freedom and confidence intervals be reported in the coefficient table.

dfpvalues specifies that parameter degrees of freedom and p-values be reported in the coefficient table.

Optimization

iterate(#) is part of StataNow. It specifies the maximum number of iterations for the APM. The default is iterate(50).

nolog is part of StataNow. It specifies that no APM iterative log be displayed.

tolerance(#) is part of StataNow. It specifies the limit for the maximum absolute difference between iterations for the projected *depvar* and *indepvars*. The default is tolerance(1e-8).

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

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

Options for MLE model

Model

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

mle requests the maximum-likelihood random-effects estimator.

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***.

Reporting

level(#); 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: iterate(#), [no]log, trace, tolerance(#), ltolerance(#), and from(*init_specs*); see [R] **Maximize**. These options are seldom used.

The following option is available with xtreg 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. For linear regression, this is the same as a random-effects estimator (both interpretations hold).

xtreg, pa is equivalent to xtgee, family(gaussian) link(id) corr(exchangeable), which are the defaults for the xtgee command. xtreg, pa allows all the relevant xtgee options such as vce(robust). Whether you use xtreg, pa or xtgee makes no difference. See [XT] **xtgee**.

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; see [XT] ***vce_options***.

rgf specifies that the robust variance estimate is multiplied by $(N-1)/(N-P)$, where N is the total number of observations and P is the number of coefficients estimated. This option can be used with family(gaussian) only when vce(robust) is either specified or implied by the use of pweights. Using this option implies that the robust variance estimate is not invariant to the scale of any weights used.

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

Reporting

`level(#)`; 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**.

Optimization

optimize_options control the iterative optimization process. These options are seldom used.

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

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

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

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

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

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

Remarks and examples

stata.com

If you have not read [XT] **xt**, please do so.

See Baltagi (2013, chap. 2) and Wooldridge (2020, chap. 14) for good overviews of fixed-effects and random-effects models. Allison (2009) provides perspective on the use of fixed- versus random-effects estimators and provides many examples using Stata.

Consider fitting models of the form

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it} \tag{1}$$

In this model, $\nu_i + \epsilon_{it}$ is the error term that we have little interest in; we want estimates of $\boldsymbol{\beta}$. ν_i is the unit-specific error term; it differs between units, but for any particular unit, its value is constant. In the pulmonary data of [XT] **xt**, a person who exercises less would presumably have a lower forced expiratory volume (FEV) year after year and so would have a negative ν_i.

ϵ_{it} is the "usual" error term with the usual properties (mean 0, uncorrelated with itself, uncorrelated with $\mathbf{x}$, uncorrelated with ν, and homoskedastic), although in a more thorough development, we could decompose $\epsilon_{it} = \upsilon_t + \omega_{it}$, assume that ω_{it} is a conventional error term, and better describe υ_t.

Before making the assumptions necessary for estimation, let's perform some useful algebra on (1). Whatever the properties of ν_i and ϵ_{it}, if (1) is true, it must also be true that

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta} + \nu_i + \overline{\epsilon}_i \tag{2}$$

where $\overline{y}_i = \sum_t y_{it}/T_i$, $\overline{\mathbf{x}}_i = \sum_t \mathbf{x}_{it}/T_i$, and $\overline{\epsilon}_i = \sum_t \epsilon_{it}/T_i$. Subtracting (2) from (1), it must be equally true that

$$(y_{it} - \overline{y}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta} + (\epsilon_{it} - \overline{\epsilon}_i) \tag{3}$$

These three equations provide the basis for estimating $\boldsymbol{\beta}$. In particular, `xtreg, fe` provides what is known as the fixed-effects estimator—also known as the within estimator—and amounts to using OLS to perform the estimation of (3). `xtreg, be` provides what is known as the between estimator and amounts to using OLS to perform the estimation of (2). `xtreg, re` provides the random-effects estimator and is a (matrix) weighted average of the estimates produced by the between and within estimators. In particular, the random-effects estimator turns out to be equivalent to estimation of

$$(y_{it} - \theta\overline{y}_i) = (1-\theta)\alpha + (\mathbf{x}_{it} - \theta\overline{\mathbf{x}}_i)\boldsymbol{\beta} + \{(1-\theta)\nu_i + (\epsilon_{it} - \theta\overline{\epsilon}_i)\} \tag{4}$$

where θ is a function of σ_ν^2 and σ_ϵ^2. If $\sigma_\nu^2 = 0$, meaning that ν_i is always 0, $\theta = 0$ and (1) can be estimated by OLS directly. Alternatively, if $\sigma_\epsilon^2 = 0$, meaning that ϵ_{it} is 0, $\theta = 1$ and the within estimator returns all the information available (which will, in fact, be a regression with an R^2 of 1).

For more reasonable cases, few assumptions are required to justify the fixed-effects estimator of (3). The estimates are, however, conditional on the sample in that the ν_i are not assumed to have a distribution but are instead treated as fixed and estimable. This statistical fine point can lead to difficulty when making out-of-sample predictions, but that aside, the fixed-effects estimator has much to recommend it.

More is required to justify the between estimator of (2), but the conditioning on the sample is not assumed because $\nu_i + \overline{\epsilon}_i$ is treated as an error term. Newly required is that we assume that ν_i and $\overline{\mathbf{x}}_i$ are uncorrelated. This follows from the assumptions of the OLS estimator but is also transparent: were ν_i and $\overline{\mathbf{x}}_i$ correlated, the estimator could not determine how much of the change in $\overline{y}_i$, associated with an increase in $\overline{\mathbf{x}}_i$, to assign to $\boldsymbol{\beta}$ versus how much to attribute to the unknown correlation. (This, of course, suggests the use of an instrumental-variable estimator, $\overline{\mathbf{z}}_i$, which is correlated with $\overline{\mathbf{x}}_i$ but uncorrelated with ν_i, though that approach is not implemented here.)

The random-effects estimator of (4) requires the same no-correlation assumption. In comparison with the between estimator, the random-effects estimator produces more efficient results, albeit ones with unknown small-sample properties. The between estimator is less efficient because it discards the over-time information in the data in favor of simple means; the random-effects estimator uses both the within and the between information.

All of this would seem to leave the between estimator of (2) with no role (except for a minor, technical part it plays in helping to estimate σ_ν^2 and σ_ϵ^2, which are used in the calculation of θ, on which the random-effects estimates depend). Let's, however, consider a variation on (1):

$$y_{it} = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta}_1 + (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta}_2 + \nu_i + \epsilon_{it} \tag{$1'$}$$

In this model, we postulate that changes in the average value of $\mathbf{x}$ for an individual have a different effect from temporary departures from the average. In an economic situation, y might be purchases of some item and $\mathbf{x}$ income; a change in average income should have more effect than a transitory change. In a clinical situation, y might be a physical response and $\mathbf{x}$ the level of a chemical in the brain; the model allows a different response to permanent rather than transitory changes.

The variations of (2) and (3) corresponding to ($1'$) are

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta}_1 + \nu_i + \overline{\epsilon}_i \tag{$2'$}$$

$$(y_{it} - \overline{y}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta}_2 + (\epsilon_{it} - \overline{\epsilon}_i) \tag{$3'$}$$

That is, the between estimator estimates $\boldsymbol{\beta}_1$ and the within $\boldsymbol{\beta}_2$, and neither estimates the other. Thus even when estimating equations like (1), it is worth comparing the within and between estimators. Differences in results can suggest models like ($1'$), or at the least some other specification error.

Finally, it is worth understanding the role of the between and within estimators with regressors that are constant over time or constant over units. Consider the model

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta}_1 + \mathbf{s}_i\boldsymbol{\beta}_2 + \mathbf{z}_t\boldsymbol{\beta}_3 + \nu_i + \epsilon_{it} \tag{1''}$$

This model is the same as (1), except that we explicitly identify the variables that vary over both time and i ($\mathbf{x}_{it}$, such as output or FEV); variables that are constant over time ($\mathbf{s}_i$, such as race or sex); and variables that vary solely over time ($\mathbf{z}_t$, such as the consumer price index or age in a cohort study). The corresponding between and within equations are

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta}_1 + \mathbf{s}_i\boldsymbol{\beta}_2 + \overline{\mathbf{z}}\boldsymbol{\beta}_3 + \nu_i + \overline{\epsilon}_i \tag{2''}$$

$$(y_{it} - \overline{y}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\boldsymbol{\beta}_1 + (\mathbf{z}_t - \overline{\mathbf{z}})\boldsymbol{\beta}_3 + (\epsilon_{it} - \overline{\epsilon}_i) \tag{3''}$$

In the between estimator of (2″), no estimate of $\boldsymbol{\beta}_3$ is possible because $\overline{\mathbf{z}}$ is a constant across the i observations; the regression-estimated intercept will be an estimate of $\alpha + \overline{\mathbf{z}}\boldsymbol{\beta}_3$. On the other hand, it can provide estimates of $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$. It can estimate effects of factors that are constant over time, such as race and sex, but to do so it must assume that ν_i is uncorrelated with those factors.

The within estimator of (3″), like the between estimator, provides an estimate of $\boldsymbol{\beta}_1$ but provides no estimate of $\boldsymbol{\beta}_2$ for time-invariant factors. Instead, it provides an estimate of $\boldsymbol{\beta}_3$, the effects of the time-varying factors. The within estimator can also provide estimates u_i for ν_i. More correctly, the estimator u_i is an estimator of $\nu_i + \mathbf{s}_i\boldsymbol{\beta}_2$. Thus u_i is an estimator of ν_i only if there are no time-invariant variables in the model. If there are time-invariant variables, u_i is an estimate of ν_i plus the effects of the time-invariant variables.

Remarks are presented under the following headings:

Assessing goodness of fit

R^2 is a popular measure of goodness of fit in ordinary regression. In our case, given $\widehat{\alpha}$ and $\widehat{\boldsymbol{\beta}}$ estimates of α and $\boldsymbol{\beta}$, we can assess the goodness of fit with respect to (1), (2), or (3). The prediction equations are, respectively,

$$\widehat{y}_{it} = \widehat{\alpha} + \mathbf{x}_{it}\widehat{\boldsymbol{\beta}} \tag{1'''}$$

$$\widehat{\overline{y}}_i = \widehat{\alpha} + \overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}} \tag{2'''}$$

$$\widehat{\widetilde{y}}_{it} = (\widehat{y}_{it} - \widehat{\overline{y}}_i) = (\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}} \tag{3'''}$$

`xtreg` reports "R-squares" corresponding to these three equations. R-squares is in quotes because the R-squares reported do not have all the properties of the OLS R^2.

The ordinary properties of R^2 include being equal to the squared correlation between $\widehat{y}$ and y and being equal to the fraction of the variation in y explained by $\widehat{y}$—formally defined as $\mathrm{Var}(\widehat{y})/\mathrm{Var}(y)$. The identity of the definitions is from a special property of the OLS estimates; in general, given a prediction $\widehat{y}$ for y, the squared correlation is not equal to the ratio of the variances, and the ratio of the variances is not required to be less than 1.

`xtreg` reports R^2 values calculated as correlations squared, calling them R^2 overall, corresponding to (1‴); R^2 between, corresponding to (2‴); and R^2 within, corresponding to (3‴). In fact, you can think of each of these three numbers as having all the properties of ordinary R^2's, if you bear in mind that the prediction being judged is not $\widehat{y}_{it}$, $\widehat{\overline{y}}_i$, and $\widehat{\widetilde{y}}_{it}$, but $\gamma_1\widehat{y}_{it}$ from the regression $y_{it} = \gamma_1\widehat{y}_{it}$; $\gamma_2\widehat{\overline{y}}_i$ from the regression $\overline{y}_i = \gamma_2\widehat{\overline{y}}_i$; and $\gamma_3\widehat{\widetilde{y}}_{it}$ from $\widetilde{y}_{it} = \gamma_3\widehat{\widetilde{y}}_{it}$.

In particular, `xtreg, be` obtains its estimates by performing OLS on (2), and therefore its reported R^2 between is an ordinary R^2. The other two reported R^2's are merely correlations squared, or, if you prefer, R^2's from the second-round regressions $y_{it} = \gamma_{11}\widehat{y}_{it}$ and $\widetilde{y}_{it} = \gamma_{13}\widehat{\widetilde{y}}_{it}$.

`xtreg, fe` obtains its estimates by performing OLS on (3), so its reported R^2 within is an ordinary R^2. As with `be`, the other R^2's are correlations squared, or, if you prefer, R^2's from the second-round regressions $\overline{y}_i = \gamma_{22}\widehat{\overline{y}}_i$ and, as with `be`, $\widetilde{y}_{it} = \gamma_{23}\widehat{\widetilde{y}}_{it}$.

`xtreg, re` obtains its estimates by performing OLS on (4); none of the R^2's corresponding to $(1''')$, $(2''')$, or $(3''')$ correspond directly to this estimator [the "relevant" R^2 is the one corresponding to (4)]. All three reported R^2's are correlations squared, or, if you prefer, from second-round regressions.

xtreg and associated commands

▷ Example 1: Between-effects model

Using `nlswork.dta` described in [XT] **xt**, we will model `ln_wage` in terms of completed years of schooling (`grade`), current age and age squared, current years worked (experience) and experience squared, current years of tenure on the current job and tenure squared, whether black (`race` = 2), whether residing in an area not designated a standard metropolitan statistical area (SMSA), and whether residing in the South.

```
. use https://www.stata-press.com/data/r18/nlswork
(National Longitudinal Survey of Young Women, 14-24 years old in 1968)
```

To obtain the between-effects estimates, we use `xtreg, be`. `nlswork.dta` has previously been `xtset idcode year` because that is what is true of the data, but for running `xtreg`, it would have been sufficient to have `xtset idcode` by itself.

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, be

Between regression (regression on group means)  Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1591                                         min =          1
     Between = 0.4900                                         avg =        6.0
     Overall = 0.3695                                         max =         15

                                                F(10,4686)        =     450.23
sd(u_i + avg(e_i.)) = .3036114                  Prob > F          =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0607602   .0020006    30.37   0.000     .0568382    .0646822
         age |   .0323158   .0087251     3.70   0.000     .0152105    .0494211
             |
 c.age#c.age |  -.0005997   .0001429    -4.20   0.000    -.0008799   -.0003194
             |
     ttl_exp |   .0138853   .0056749     2.45   0.014     .0027598    .0250108
             |
   c.ttl_exp#|
   c.ttl_exp |   .0007342   .0003267     2.25   0.025     .0000936    .0013747
             |
      tenure |   .0698419   .0060729    11.50   0.000     .0579361    .0817476
             |
    c.tenure#|
    c.tenure |  -.0028756   .0004098    -7.02   0.000    -.0036789   -.0020722
             |
        race |
      Black  |  -.0564167   .0105131    -5.37   0.000    -.0770272   -.0358061
    not_smsa |  -.1860406   .0112495   -16.54   0.000    -.2080949   -.1639862
       south |  -.0993378    .010136    -9.80   0.000    -.1192091   -.0794665
       _cons |   .3339113   .1210434     2.76   0.006     .0966093    .5712133
------------------------------------------------------------------------------
```

The between-effects regression is estimated on person-averages, so the "n = 4697" result is relevant. xtreg, be reports the "number of observations" and group-size information: describe in [XT] xt showed that we have 28,534 "observations"—person-years, really—of data. If we take the subsample that has no missing values in ln_wage, grade, ..., south leaves us with 28,091 observations on person-years, reflecting 4,697 persons, each observed for an average of 6.0 years.

For goodness of fit, the R^2 between is directly relevant; our R^2 is 0.4900. If, however, we use these estimates to predict the within model, we have an R^2 of 0.1591. If we use these estimates to fit the overall data, our R^2 is 0.3695.

The F statistic tests that the coefficients on the regressors grade, age, ..., south are all jointly zero. Our model is significant.

The root mean squared error of the fitted regression, which is an estimate of the standard deviation of $\nu_i + \overline{\epsilon}_i$, is 0.3036.

For our coefficients, each year of schooling increases hourly wages by 6.1%; age increases wages up to age 26.9 and thereafter decreases them (because the quadratic $ax^2 + bx + c$ turns over at $x = -b/2a$, which for our age and c.age#c.age coefficients is $0.0323158/(2 \times 0.0005997) \approx 26.9$); total experience increases wages at an increasing rate (which is surprising and bothersome); tenure on the current job increases wages up to a tenure of 12.1 years and thereafter decreases them; wages of blacks are, these things held constant, (approximately) 5.6% below that of nonblacks (approximately because 2.race is an indicator variable); residing in a non-SMSA (rural area) reduces wages by 18.6%; and residing in the South reduces wages by 9.9%.

◁

▷ Example 2: Fixed-effects model

To fit the same model with the fixed-effects estimator, we specify the fe option.

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, fe
note: grade omitted because of collinearity.
note: 2.race omitted because of collinearity.

Fixed-effects (within) regression               Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1727                                         min =          1
     Between = 0.3505                                         avg =        6.0
     Overall = 0.2625                                         max =         15

                                                F(8, 23386)       =     610.12
corr(u_i, Xb) = 0.1936                          Prob > F          =     0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |          0  (omitted)
         age |   .0359987   .0033864    10.63   0.000     .0293611    .0426362
             |
 c.age#c.age |   -.000723   .0000533   -13.58   0.000    -.0008274   -.0006186
             |
     ttl_exp |   .0334668   .0029653    11.29   0.000     .0276545     .039279
             |
   c.ttl_exp#|
   c.ttl_exp |   .0002163   .0001277     1.69   0.090    -.0000341    .0004666
             |
      tenure |   .0357539   .0018487    19.34   0.000     .0321303    .0393775
             |
    c.tenure#|
    c.tenure |  -.0019701    .000125   -15.76   0.000    -.0022151   -.0017251
             |
        race |
      Black  |          0  (omitted)
    not_smsa |  -.0890108   .0095316    -9.34   0.000    -.1076933   -.0703282
       south |  -.0606309   .0109319    -5.55   0.000    -.0820582   -.0392036
       _cons |    1.03732   .0485546    21.36   0.000     .9421496     1.13249
-------------+----------------------------------------------------------------
     sigma_u |  .35562203
     sigma_e |  .29068923
         rho |  .59946283   (fraction of variance due to u_i)
------------------------------------------------------------------------------
F test that all u_i=0: F(4696, 23386) = 6.65                 Prob > F = 0.0000
```

The observation summary at the top is the same as for the between-effects model, although this time it is the "Number of obs" that is relevant.

Our three R^2's are not too different from those reported previously; the R^2 within is slightly higher (0.1727 versus 0.1591), and the R^2 between is a little lower (0.3505 versus 0.4900), as expected, because the between estimator maximizes R^2 between and the within estimator R^2 within. In terms of overall fit, these estimates are somewhat worse (0.2625 versus 0.3695).

If the unobserved time-invariant component ν is not correlated with the regressors, estimates from the fixed-effects model are consistent but inefficient relative to estimates from the random-effects model. In this case, the interpretation of `sigma_u` in the coefficient table is the same for the fixed-effects and random-effects models. However, `sigma_u` is a nuisance parameter when ν is correlated with the covariates.

Here both `grade` and `2.race` were omitted from the model because they do not vary over time. Our estimate u_i is an estimate of ν_i plus the effects of any time-invariant variables, including `2.grade` and `2.race`, so our estimate of the standard deviation is based on the variation in ν_i, `grade`, `2.race`, and any other time-invariant variables. On the other hand, had `2.race` and `grade` been omitted merely because they were collinear with the other regressors in our model, u_i would be an estimate of ν_i, and 0.355622 would be an estimate of σ_ν. (`xtsum` and `xttab` allow you to determine whether a variable is time invariant; see [XT] **xtsum** and [XT] **xttab**.)

Regardless of the status of u_i, our estimate of the standard deviation of ϵ_{it} is valid (and, in fact, is the estimate that would be used by the random-effects estimator to produce its results).

Our estimate of the correlation of u_i with $\mathbf{x}_{it}$ suffers from the problem of what u_i measures. We find correlation but cannot say whether this is correlation of ν_i with $\mathbf{x}_{it}$ or merely correlation of `grade` and `2.race` with $\mathbf{x}_{it}$. In any case, the fixed-effects estimator is robust to such a correlation, and the other estimates it produces are unbiased.

So, although this estimator produces no estimates of the effects of `grade` and `2.race`, it does predict that age has a positive effect on wages up to age 24.9 years (compared with 26.9 years estimated by the between estimator); that total experience still increases wages at an increasing rate (which is still bothersome); that tenure increases wages up to 9.1 years (compared with 12.1); that living in a non-SMSA reduces wages by 8.9% (compared with a more drastic 18.6%); and that living in the South reduces wages by 6.1% (as compared with 9.9%).

◁

▷ Example 3: Fixed-effects models with robust standard errors

If we suspect that there is heteroskedasticity or within-panel serial correlation in the idiosyncratic error term ϵ_{it}, we could specify the `vce(robust)` or the `vce(hc2)` option:

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, fe vce(robust)
note: grade omitted because of collinearity.
note: 2.race omitted because of collinearity.

Fixed-effects (within) regression               Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1727                                         min =          1
     Between = 0.3505                                         avg =        6.0
     Overall = 0.2625                                         max =         15

                                                F(8, 4696)        =     273.86
corr(u_i, Xb) = 0.1936                          Prob > F          =     0.0000

                           (Std. err. adjusted for 4,697 clusters in idcode)

                           Robust
     ln_wage  Coefficient  std. err.      t    P>|t|     [95% conf. interval]

       grade            0  (omitted)
         age     .0359987   .0052407     6.87   0.000     .0257243     .046273

 c.age#c.age     -.000723   .0000845    -8.56   0.000    -.0008887   -.0005573

     ttl_exp     .0334668    .004069     8.22   0.000     .0254896    .0414439

   c.ttl_exp#
   c.ttl_exp     .0002163   .0001763     1.23   0.220    -.0001294    .0005619

      tenure     .0357539   .0024683    14.49   0.000     .0309148     .040593

    c.tenure#
    c.tenure    -.0019701   .0001696   -11.62   0.000    -.0023026   -.0016376

        race
      Black             0  (omitted)
    not_smsa    -.0890108   .0137629    -6.47   0.000    -.1159926    -.062029
       south    -.0606309   .0163366    -3.71   0.000    -.0926583   -.0286035
       _cons      1.03732   .0739644    14.02   0.000     .8923149    1.182325

     sigma_u    .35562203
     sigma_e    .29068923
         rho    .59946283   (fraction of variance due to u_i)
```

Although the estimated coefficients are the same with and without the `vce(robust)` option, the robust estimator produced larger standard errors and a p-value for `c.ttl_exp#c.ttl_exp` above the conventional 10%. The F test of $\nu_i = 0$ is suppressed because it is too difficult to compute the robust form of the statistic when there are more than a few panels.

◁

❑ Technical note

The robust standard errors reported above are identical to those obtained by clustering on the panel variable `idcode`. Clustering on the panel variable produces an estimator of the VCE that is robust to cross-sectional heteroskedasticity and within-panel (serial) correlation that is asymptotically equivalent to that proposed by Arellano (1987). Although the example above applies the fixed-effects estimator, the robust and cluster–robust VCE estimators are also available for the random-effects estimator. Wooldridge (2020) and Arellano (2003) discuss these robust and cluster–robust VCE estimators for the fixed-effects and random-effects estimators. More details are available in *Methods and formulas*.

❑

▷ Example 4: Random-effects model

Refitting our log-wage model with the random-effects estimator, we obtain

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, re theta

Random-effects GLS regression                   Number of obs     =     28,091
Group variable: idcode                          Number of groups  =      4,697

R-squared:                                      Obs per group:
     Within  = 0.1715                                         min =          1
     Between = 0.4784                                         avg =        6.0
     Overall = 0.3708                                         max =         15

                                                Wald chi2(10)     =    9244.74
corr(u_i, X) = 0 (assumed)                      Prob > chi2       =     0.0000

------------------- theta --------------------
  min      5%       median        95%      max
0.2520   0.2520     0.5499     0.7016   0.7206
```

ln_wage	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
grade	.0646499	.0017812	36.30	0.000	.0611589	.0681409
age	.0368059	.0031195	11.80	0.000	.0306918	.0429201
c.age#c.age	-.0007133	.00005	-14.27	0.000	-.0008113	-.0006153
ttl_exp	.0290208	.002422	11.98	0.000	.0242739	.0337678
c.ttl_exp#c.ttl_exp	.0003049	.0001162	2.62	0.009	.000077	.0005327
tenure	.0392519	.0017554	22.36	0.000	.0358113	.0426925
c.tenure#c.tenure	-.0020035	.0001193	-16.80	0.000	-.0022373	-.0017697
race						
Black	-.053053	.0099926	-5.31	0.000	-.0726381	-.0334679
not_smsa	-.1308252	.0071751	-18.23	0.000	-.1448881	-.1167622
south	-.0868922	.0073032	-11.90	0.000	-.1012062	-.0725781
_cons	.2387207	.049469	4.83	0.000	.1417633	.3356781
sigma_u	.25790526					
sigma_e	.29068923					
rho	.44045273	(fraction of variance due to u_i)				

According to the R^2's, this estimator performs worse within than the within fixed-effects estimator and worse between than the between estimator, as it must, and slightly better overall.

We estimate that σ_ν is 0.2579 and σ_ϵ is 0.2907 and, by assertion, assume that the correlation of ν and $\mathbf{x}$ is zero.

All that is known about the random-effects estimator is its asymptotic properties, so rather than reporting an F statistic for overall significance, `xtreg, re` reports a χ^2. Taken jointly, our coefficients are significant.

`xtreg, re` also reports a summary of the distribution of θ_i, an ingredient in the estimation of (4). θ is not a constant here because we observe women for unequal periods.

We estimate that schooling has a rate of return of 6.5% (compared with 6.1% between and no estimate within); that the increase of wages with age turns around at 25.8 years (compared with 26.9 between and 24.9 within); that total experience yet again increases wages increasingly; that the effect of job tenure turns around at 9.8 years (compared with 12.1 between and 9.1 within); that being black reduces wages by 5.3% (compared with 5.6% between and no estimate within); that living in a non-SMSA reduces wages 13.1% (compared with 18.6% between and 8.9% within); and that living in the South reduces wages 8.7% (compared with 9.9% between and 6.1% within).

◁

▷ Example 5: Random-effects model fit using ML

We could also have fit this random-effects model with the maximum likelihood estimator:

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, mle
Fitting constant-only model:
Iteration 0:  Log likelihood = -12663.954
Iteration 1:  Log likelihood = -12649.756
Iteration 2:  Log likelihood = -12649.614
Iteration 3:  Log likelihood = -12649.614
Fitting full model:
Iteration 0:  Log likelihood =  -8922.145
Iteration 1:  Log likelihood = -8853.6409
Iteration 2:  Log likelihood = -8853.4255
Iteration 3:  Log likelihood = -8853.4254
Random-effects ML regression                    Number of obs     =  28,091
Group variable: idcode                          Number of groups =   4,697
Random effects u_i ~ Gaussian                   Obs per group:
                                                             min =       1
                                                             avg =     6.0
                                                             max =      15
                                                LR chi2(10)       = 7592.38
Log likelihood = -8853.4254                     Prob > chi2       =  0.0000

------------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       grade |   .0646093   .0017372    37.19   0.000     .0612044    .0680142
         age |   .0368531   .0031226    11.80   0.000      .030733    .0429732
             |
 c.age#c.age |  -.0007132   .0000501   -14.24   0.000    -.0008113    -.000615
             |
     ttl_exp |   .0288196   .0024143    11.94   0.000     .0240877    .0335515
             |
   c.ttl_exp#|
   c.ttl_exp |    .000309   .0001163     2.66   0.008     .0000811    .0005369
             |
      tenure |   .0394371   .0017604    22.40   0.000     .0359868    .0428875
             |
    c.tenure#|
    c.tenure |  -.0020052   .0001195   -16.77   0.000    -.0022395   -.0017709
             |
        race |
      Black  |  -.0533394   .0097338    -5.48   0.000    -.0724172   -.0342615
    not_smsa |  -.1323433   .0071322   -18.56   0.000    -.1463221   -.1183644
       south |  -.0875599   .0072143   -12.14   0.000    -.1016998   -.0734201
       _cons |   .2390837   .0491902     4.86   0.000     .1426727    .3354947
-------------+----------------------------------------------------------------
    /sigma_u |   .2485556   .0035017                      .2417863    .2555144
    /sigma_e |   .2918458    .001352                       .289208    .2945076
         rho |   .4204033   .0074828                      .4057959    .4351212
------------------------------------------------------------------------------
LR test of sigma_u=0: chibar2(01) = 7339.84           Prob >= chibar2 = 0.000
```

The estimates are nearly the same as those produced by `xtreg, re`—the GLS estimator. For instance, `xtreg, re` estimated the coefficient on `grade` to be 0.0646499, `xtreg, mle` estimated 0.0646093, and the ratio is $0.0646499/0.0646093 = 1.001$ to three decimal places. Similarly, the standard errors are nearly equal: $0.0017811/0.0017372 = 1.025$. Below we compare all 11 coefficients:

	Coefficient ratio			SE ratio		
Estimator	mean	min.	max.	mean	min.	max.
`xtreg, mle` (ML)	1.	1.	1.	1.	1.	1.
`xtreg, re` (GLS)	.997	.987	1.007	1.006	.997	1.027

◁

▷ Example 6: Population-averaged model

We could also have fit this model with the population-averaged estimator:

```
. xtreg ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure 2.race not_smsa south, pa
Iteration 1:  Tolerance = .0310561
Iteration 2:  Tolerance = .00074898
Iteration 3:  Tolerance = .0000147
Iteration 4:  Tolerance = 2.880e-07
GEE population-averaged model                    Number of obs     =  28,091
Group variable: idcode                           Number of groups =   4,697
Family: Gaussian                                 Obs per group:
Link:   Identity                                              min =       1
Correlation: exchangeable                                     avg =     6.0
                                                              max =      15
                                                 Wald chi2(10)     = 9598.89
Scale parameter = .1436709                       Prob > chi2       =  0.0000

---------------------------------------------------------------------------
     ln_wage | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+-------------------------------------------------------------
       grade |   .0645427   .0016829    38.35   0.000     .0612442    .0678412
         age |    .036932   .0031509    11.72   0.000     .0307564    .0431076
             |
 c.age#c.age |  -.0007129   .0000506   -14.10   0.000    -.0008121   -.0006138
             |
     ttl_exp |   .0284878   .0024169    11.79   0.000     .0237508    .0332248
             |
   c.ttl_exp#|
   c.ttl_exp |   .0003158   .0001172     2.69   0.007      .000086    .0005456
             |
      tenure |   .0397468   .0017779    22.36   0.000     .0362621    .0432315
             |
    c.tenure#|
    c.tenure |   -.002008   .0001209   -16.61   0.000    -.0022449   -.0017711
             |
        race |
       Black |  -.0538314   .0094086    -5.72   0.000     -.072272   -.0353909
    not_smsa |  -.1347788   .0070543   -19.11   0.000    -.1486049   -.1209526
       south |  -.0885969   .0071132   -12.46   0.000    -.1025386   -.0746552
       _cons |   .2396286   .0491465     4.88   0.000     .1433034    .3359539
---------------------------------------------------------------------------
```

These results differ from those produced by `xtreg, re` and `xtreg, mle`. Coefficients are larger and standard errors smaller. `xtreg, pa` is simply another way to run the `xtgee` command. That is, we would have obtained the same output had we typed

```
. xtgee ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race not_smsa south
```
(*output omitted because it is the same as above*)

See [XT] **xtgee**. In the language of `xtgee`, the random-effects model corresponds to an `exchangeable` correlation structure and `identity` link, and `xtgee` also allows other correlation structures. Let's stay with the random-effects model, however. `xtgee` will also produce robust estimates of variance, and we refit this model that way by typing

```
. xtgee ln_w grade age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp
> tenure c.tenure#c.tenure 2.race not_smsa south, vce(robust)
```
(*output omitted, coefficients the same, standard errors different*)

In the previous example, we presented a table comparing `xtreg, re` with `xtreg, mle`. Below we add the results from the estimates shown and the ones we did with `xtgee, vce(robust)`:

Estimator		Coefficient ratio mean	min.	max.	SE ratio mean	min.	max.
`xtreg, mle`	(ML)	1.	1.	1.	1.	1.	1.
`xtreg, re`	(GLS)	.997	.987	1.007	1.006	.997	1.027
`xtreg, pa`	(PA)	1.060	.847	1.317	.853	.626	.986
`xtgee, vce(robust)`	(PA)	1.060	.847	1.317	1.306	.957	1.545

So, which are right? This is a real dataset, and we do not know. However, in example 2 in [XT] **xtreg postestimation**, we will present evidence that the assumptions underlying the `xtreg, re` and `xtreg, mle` results are not met.

◁

▷ Example 7: Controlling for high-dimensional categorical variables (StataNow)

Let's assume that in the labor market model we have been fitting, we would like to account for county effects. The United States, as of 2024, had approximately 3,200 counties. To fit a fixed-effects model, we could type

```
. xtreg ... i.county, fe
```

However, this would add the 3,200 county indicators as regressors in our model. If we are not interested in the coefficients on the county indicators, this creates an unnecessary computational burden. We can fit the model in a more efficient way by absorbing the county effects. To do this, we type

```
. xtreg ..., fe absorb(county)
```

Now, instead of estimating 3,200 coefficients, we absorb the county effects as well as the panel effects.

For illustrative purposes, we generate a county variable that mimics the description above.

```
. set seed 123
. generate county = runiformint(1,3200)
```

Then we fit the model.

```
. set seed 123
. generate county = runiformint(1,3200)
```

Below the command, we see iteration logs showing the maximum absolute difference (MAD) for the APM, which is the algorithm used to absorb the effects of panel and county variables. The smaller the MAD, the closer the APM is to the projection of the model fit by including indicator variables for all levels of `county`. Thus, the MAD should be as close to 0 as possible to absorb county and panel effects and obtain regression coefficients for the other predictors.

Following the iteration logs, we obtain output similar to the output from `xtreg, fe` when the `absorb()` option was not specified. However, we see an additional table that reports which variables were absorbed and the number of categories in each of the variables. No coefficients are reported for the counties. The county effects are absorbed.

Note that this output includes an estimate of σ_u and an F test that all u_is are 0; each of these requires additional computation time. If you are not interested in the estimate of σ_u, you can specify the `nosigmau` option. If you are not interested in σ_u or the F test that the u_is are 0, you can specify the `nouitest` option.

```
. xtreg ln_w age c.age#c.age ttl_exp c.ttl_exp#c.ttl_exp tenure
> c.tenure#c.tenure not_smsa south, fe
> absorb(county) nouitest
```

(output omitted)

```
Fixed-effects (within) regression               Number of obs     =     28,093
Group variable: idcode                          Number of groups  =      4,699

R-squared:                                      Obs per group:
     Within  = 0.2881                                         min =          1
     Between = 0.3479                                         avg =        6.0
     Overall = 0.2604                                         max =         15

                                                F(8, 20187)       =     533.12
```

Absorbed variable	Levels
county	3,200

ln_wage	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
age	.0359171	.0036425	9.86	0.000	.0287774	.0430567
c.age#c.age	-.0007245	.0000573	-12.65	0.000	-.0008368	-.0006122
ttl_exp	.0343766	.0031888	10.78	0.000	.0281262	.040627
c.ttl_exp# c.ttl_exp	.0001783	.0001374	1.30	0.194	-.0000911	.0004477
tenure	.0356387	.0019824	17.98	0.000	.0317531	.0395244
c.tenure# c.tenure	-.0019394	.0001339	-14.48	0.000	-.0022019	-.0016769
not_smsa	-.0909885	.0102319	-8.89	0.000	-.1110438	-.0709331
south	-.0514181	.0117182	-4.39	0.000	-.0743867	-.0284494
_cons	1.034046	.0522015	19.81	0.000	.9317273	1.136366
sigma_e	.29022084					

Specifying `nouitest` can be particularly helpful in improving speed when absorbing categorical variables with many levels.

◁

Stored results

xtreg, re 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(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(Tcon)	1 if T is constant
e(sigma)	square root of the sum of e(sigma_e)^2 and e(sigma_u)^2
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of ϵ_{it}
e(r2_w)	R^2 for within model
e(r2_o)	R^2 for overall model
e(r2_b)	R^2 for between model
e(N_clust)	number of clusters
e(chi2)	χ^2
e(p)	p-value for model test
e(rho)	ρ
e(thta_min)	minimum θ
e(thta_5)	θ, 5th percentile
e(thta_50)	θ, 50th percentile
e(thta_95)	θ, 95th percentile
e(thta_max)	maximum θ
e(rmse)	root mean squared error of GLS regression
e(Tbar)	harmonic mean of group sizes
e(rank)	rank of e(V)
Macros	
e(cmd)	xtreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(model)	re
e(clustvar)	name of cluster 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(sa)	sa, 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(bf)	coefficient vector for fixed-effects model
e(theta)	θ
e(V)	variance–covariance matrix of the estimators
e(VCEf)	VCE for fixed-effects model
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtreg, be` stores the following in `e()`:

Scalars	
`e(N)`	number of observations
`e(N_g)`	number of groups
`e(mss)`	model sum of squares
`e(df_m)`	model degrees of freedom
`e(rss)`	residual sum of squares
`e(df_r)`	residual degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(Tcon)`	`1` if T is constant
`e(r2)`	R^2
`e(r2_a)`	adjusted R^2
`e(r2_w)`	R^2 for within model
`e(r2_o)`	R^2 for overall model
`e(r2_b)`	R^2 for between model
`e(F)`	F statistic
`e(p)`	p-value for model test
`e(rmse)`	root mean squared error
`e(Tbar)`	harmonic mean of group sizes
`e(rank)`	rank of `e(V)`
Macros	
`e(cmd)`	`xtreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`be`
`e(typ)`	`WLS`, if `wls` specified
`e(title)`	title in estimation output
`e(vce)`	*vcetype* specified in `vce()`
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(marginsok)`	predictions allowed by `margins`
`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(V)`	variance–covariance matrix of the estimators
Functions	
`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtreg, fe stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(mss)	model sum of squares
e(df_m)	model degrees of freedom
e(rss)	residual sum of squares
e(df_r)	residual degrees of freedom
e(tss)	total sum of squares
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(Tcon)	1 if T is constant
e(corr)	corr(u_i, Xb)
e(sigma)	square root of the sum of e(sigma_e)^2 and e(sigma_u)^2
e(sigma_u)	panel-level standard deviation
e(sigma_e)	standard deviation of ϵ_{it}
e(r2)	R^2
e(r2_a)	adjusted R^2
e(r2_w)	R^2 for within model
e(r2_o)	R^2 for overall model
e(r2_b)	R^2 for between model
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(N_clust)	number of clusters
e(k_absorb)	total number of absorbed categories, including those of the panel variable
e(rho)	ρ
e(F)	F statistic
e(F_f)	F statistic for test of u_i=0
e(p)	p-value for model test
e(p_f)	p-value for test of u_i=0
e(df_a)	degrees of freedom for absorbed panel effect
e(df_b)	numerator degrees of freedom for F statistic
e(rmse)	root mean squared error
e(Tbar)	mean of group sizes
e(rank)	rank of e(V)
e(converged)	1 if APM converged, 0 otherwise

Macros

e(cmd)	xtreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivar)	variable denoting groups
e(absvar)	names of absorbed variables, including the panel variable
e(apm)	alternating projection method
e(model)	fe
e(wtype)	weight type
e(wexp)	weight expression
e(clustvar)	names of cluster variables
e(cluster#)	cluster combination #
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
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(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance
`e(adj_df)`	adjusted degrees of freedom when `vce(hc2, dfadjust)` is specified
`e(kcluster)`	cluster sizes, multiway clustering
`e(kabsorb)`	number of levels for each absorbed variable, including the panel variable

Functions

`e(sample)`	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

`xtreg, mle` 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(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`e(sigma_u)`	panel-level standard deviation
`e(sigma_e)`	standard deviation of ϵ_{it}
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(ll_c)`	log likelihood, comparison model
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(chi2_c)`	χ^2 for comparison test
`e(p)`	p-value for model test
`e(rho)`	ρ
`e(rank)`	rank of `e(V)`

Macros

`e(cmd)`	`xtreg`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(ivar)`	variable denoting groups
`e(model)`	`ml`
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`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(distrib)`	`Gaussian`; the distribution of the RE
`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(V)	variance–covariance matrix of the estimators
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

xtreg, 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)	xtreg
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)	Gaussian
e(link)	identity; link function
e(corr)	correlation structure
e(scale)	x2, dev, phi, or #; scale parameter
e(wtype)	weight type
e(wexp)	weight expression
e(offset)	linear offset variable
e(chi2type)	Wald; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(rgf)	rgf, if rgf specified
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

In addition to the above, the following is stored in `r()`:

Matrices

`r(table)`	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

Methods and formulas

The model to be fit is

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \nu_i + \epsilon_{it}$$

for $i = 1, \dots, n$ and, for each i, $t = 1, \dots, T$, of which T_i periods are actually observed.

Methods and formulas are presented under the following headings:

xtreg, fe

`xtreg, fe` produces estimates by running OLS on

$$(y_{it} - \overline{y}_i + \overline{\overline{y}}) = \alpha + (\mathbf{x}_{it} - \overline{\mathbf{x}}_i + \overline{\overline{\mathbf{x}}})\boldsymbol{\beta} + (\epsilon_{it} - \overline{\epsilon}_i + \overline{\nu}) + \overline{\overline{\epsilon}}$$

where $\overline{y}_i = \sum_{t=1}^{T_i} y_{it}/T_i$, and similarly, $\overline{\overline{y}} = \sum_i \sum_t y_{it}/(nT_i)$. The conventional covariance matrix of the estimators is adjusted for the extra $n-1$ estimated means, so results are the same as using OLS on (1) to estimate ν_i directly. 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 (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), 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 induced by the within transform. The panel-nesting restriction is also enforced for multiway clustering, `vce(cluster` *clustvarlist*`)`, which includes all combinations of the cluster variables. See *Methods and formulas* in [R] **regress** for computational details of multiway clustering.

`xtreg, fe` with `vce(hc2` [*clustvar*][`, dfadjust`]`)` specifies an alternative bias correction for the cluster–robust variance calculation. If *clustvar* is not specified, *panelvar* is used in the VCE computation. See *Robust calculation for regress* in *Methods and formulas* in [R] **regress** for a description of the VCE produced by `vce(hc2)` and the adjusted degrees-of-freedom computations.

From the estimates $\widehat{\alpha}$ and $\widehat{\boldsymbol{\beta}}$, estimates u_i of ν_i are obtained as $u_i = \overline{y}_i - \widehat{\alpha} - \overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}$. Reported from the calculated u_i are its standard deviation and its correlation with $\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}$. Reported as the standard deviation of e_{it} is the regression's estimated root mean squared error, s, which is adjusted (as previously stated) for the $n-1$ estimated means.

Reported as R^2 within is the R^2 from the mean-deviated regression.

Reported as R^2 between is $\text{corr}(\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}, \overline{y}_i)^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

Absorbed variables with xtreg, fe (StataNow)

Suppose we have h categorical variables, $C_1, \dots, C_h$, that we would like to include as controls in our fixed-effects regression. Let m_k be the number of categories for variable C_k, and let $\mathbf{d}_{k(it)}$ be the $1 \times m_k$ indicator vector for panel i at time t. Thus, vector $\mathbf{d}_{k(it)}$ is a unit vector with a 1 in the category of variable C_k in panel i at time t, and 0 everywhere else. Let $\mathbf{D}_{k(i)}$ denote the $T_i \times m_k$ matrix of indicators for variable C_k and panel i:

$$\mathbf{D}_{k(i)} = \begin{pmatrix} \mathbf{d}_{k(i1)} \\ \mathbf{d}_{k(i2)} \\ \vdots \\ \mathbf{d}_{k(iT_i)} \end{pmatrix}$$

We would like to fit the model

$$y_{it} = \alpha + \mathbf{x}_{it}\boldsymbol{\beta} + \mathbf{d}_{1(it)}\gamma_1 + \cdots + \mathbf{d}_{h(it)}\gamma_h + \nu_i + \epsilon_{it}$$

for $t = 1, \dots, T_i$, and $i = 1, \dots, n$.

Estimating the $\boldsymbol{\gamma}$ coefficients in the same manner as the $\boldsymbol{\beta}$ coefficients can be computationally expensive if the categorical variables have many levels. Instead, we absorb the effects of these categorical variables along with the panel effects.

When absorbing categorical variables, regression coefficients are computed by first running an APM and then applying Frisch–Waugh–Lovell's theorem (see Hansen [2022, 82]). The APM is used to obtain the residuals of y_{it} and $\mathbf{x}_{it}$ after regressing them (separately) on all the categorical variables. The speed of the APM is further improved by using a conjugate gradient acceleration technique (Hernández-Ramos, Escalante, and Raydan 2011).

Two projection methods are available: Halperin (the default) and Cimmino. You can specify which to use with the `halperin` or `cimmino` suboption within the `absorb()` option. We describe both methods below.

Let $\mathbf{D}_k$ be the $N \times m_k$ matrix of indicators for the kth absorbed variable (where $N = \sum_i^n T_i$):

$$\mathbf{D}_k = \begin{pmatrix} \mathbf{D}_{k(1)} \\ \mathbf{D}_{k(2)} \\ \vdots \\ \mathbf{D}_{k(n)} \end{pmatrix}$$

The orthonormal projection matrix for C_k is therefore given by $\mathbf{P}_k = \mathbf{D}_k(\mathbf{D}_k'\mathbf{D}_k)^{-1}\mathbf{D}_k'$. By convention, we define the first absorbed variable ($k = 1$) as the panel variable. Let $\mathbf{y}$ be the $N \times 1$ vector with the values of the dependent variable in the sample. Similarly, let $\mathbf{X}$ be the matrix with the values of the covariates. Thus, the product $\overline{\mathbf{y}}_k = \mathbf{P}_k\mathbf{y}$ is the projection of the dependent variable onto the column space of $\mathbf{D}_k$. That is, $\overline{\mathbf{y}}_k$ is the $N \times 1$ vector containing the (repeated) means of y_{it} for each level of C_k, in the order that these levels appear in the sample. The same projection can be applied to the columns of covariate matrix $\mathbf{X}$, that is, $\overline{\mathbf{X}}_k = \mathbf{P}_k\mathbf{X}$.

The Halperin algorithm first sets $\widetilde{\mathbf{y}}_0^{(1)} = \mathbf{y}$ and loops over the h absorbed variables computing projection residuals $\widetilde{\mathbf{y}}_k^{(1)} = (\mathbf{I} - \mathbf{P_k})\widetilde{\mathbf{y}}_{k-1}^{(1)}$, for $k = 1, \dots, h$. Then, it repeats the loop with $\widetilde{\mathbf{y}}_0^{(j+1)} = \widetilde{\mathbf{y}}_h^{(j)}$ until convergence at $j = \jmath$. Convergence is declared when $|\widetilde{\mathbf{y}}_h^{(\jmath)} - \widetilde{\mathbf{y}}_0^{(\jmath)}|$ is less than the specified tolerance, where the matrix norm $|\cdot|$ is defined as the largest entry in absolute value. The same computations are applied to the columns of covariate matrix $\mathbf{X}$. On convergence, the overall mean $\overline{\overline{y}}$ is added to $\widetilde{\mathbf{y}}_h^{(\jmath)}$, and the vector of means $\overline{\overline{\mathbf{x}}}$ is added to the columns of $\widetilde{\mathbf{X}}_h^{(\jmath)}$.

The Cimmino algorithm first sets $\widetilde{\mathbf{y}}^{(0)} = \mathbf{y}$ and then iteratively computes

$$\widetilde{\mathbf{y}}^{(j+1)} = \left(\mathbf{I} - \frac{1}{h}\sum_{k=1}^{h}\mathbf{P}_k\right)\widetilde{\mathbf{y}}^{(j)}$$

until convergence at $j + 1 = \jmath$. It then repeats the process for covariance matrix $\mathbf{X}$ and adds the overall means, $\overline{\overline{y}}$ and $\overline{\overline{\mathbf{x}}}$, to $\widetilde{\mathbf{y}}^{(\jmath)}$ and to the columns of $\widetilde{\mathbf{X}}^{(\jmath)}$, respectively.

Efficient computation of the projection $\mathbf{P}\mathbf{y}$ can be done in Mata without generating the $N \times N$ matrix $\mathbf{P}$ directly (see [M-5] **panelsum()**, for instance).

An estimate of σ_u, the standard deviation of the panel effects, is obtained by applying the APM algorithm to $\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$. A discussion of estimating the panel effects can be found in *Remarks and examples* of [XT] **xtreg postestimation**.

For a discussion of controlling for high-dimensional categorical variables in linear and nonlinear models, see Correia (2016) and Correia, Guimarães, and Zylkin (2020).

xtreg, be

`xtreg, be` fits the following model:

$$\overline{y}_i = \alpha + \overline{\mathbf{x}}_i\boldsymbol{\beta} + \nu_i + \overline{\epsilon}_i$$

Estimation is via OLS unless T_i is not constant and the `wls` option is specified. Otherwise, the estimation is performed via WLS. The estimates and conventional VCE are obtained from `regress` for both cases, but for WLS, `[aweight=`T_i`]` is specified.

Reported as R^2 between is the R^2 from the fitted regression.

Reported as R^2 within is $\text{corr}\big\{(\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}}, y_{it} - \overline{y}_i\big\}^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

xtreg, re

The key to the random-effects estimator is the GLS transform. Given estimates of the idiosyncratic component, $\widehat{\sigma}_e^2$, and the individual component, $\widehat{\sigma}_u^2$, the GLS transform of a variable z for the random-effects model is

$$z_{it}^* = z_{it} - \widehat{\theta}_i \overline{z}_i$$

where $\overline{z}_i = 1/T_i \sum_{t=1}^{T_i} z_{it}$ and

$$\widehat{\theta}_i = 1 - \sqrt{\frac{\widehat{\sigma}_e^2}{T_i \widehat{\sigma}_u^2 + \widehat{\sigma}_e^2}}$$

Given an estimate of $\widehat{\theta}_i$, one transforms the dependent and independent variables, and then the coefficient estimates and the conventional variance–covariance matrix come from an OLS regression of y_{it}^* on $\mathbf{x}_{it}^*$ and the transformed constant $1 - \widehat{\theta}_i$. 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**; in particular, see *Introduction* and *Methods and formulas*. Wooldridge (2020) and Arellano (2003) discuss this application of the Huber/White/sandwich VCE estimator. As discussed by Wooldridge (2020), 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.

Stata has two implementations of the Swamy–Arora method for estimating the variance components. They produce the same results in balanced panels and share the same estimator of σ_e^2. However, the two methods differ in their estimator of σ_u^2 in unbalanced panels. We call the first $\widehat{\sigma}_{u\overline{T}}^2$ and the second $\widehat{\sigma}_{\mathrm{uSA}}^2$. Both estimators are consistent; however, $\widehat{\sigma}_{\mathrm{uSA}}^2$ has a more elaborate adjustment for small samples than $\widehat{\sigma}_{u\overline{T}}^2$. (See Baltagi [2013], Baltagi and Chang [1994], and Swamy and Arora [1972] for derivations of these methods.)

Both methods use the same function of within residuals to estimate the idiosyncratic error component σ_e. Specifically,

$$\widehat{\sigma}_e^2 = \frac{\sum_{i=1}^{n} \sum_{t=1}^{T_i} e_{it}^2}{N - n - K + 1}$$

where

$$e_{it} = (y_{it} - \overline{y}_i + \overline{\overline{y}}) - \widehat{\alpha}_w - (\mathbf{x}_{it} - \overline{\mathbf{x}}_i + \overline{\overline{\mathbf{x}}})\widehat{\boldsymbol{\beta}}_w$$

and $\widehat{\alpha}_w$ and $\widehat{\boldsymbol{\beta}}_w$ are the within estimates of the coefficients and $N = \sum_{i=1}^{n} T_i$. After passing the within residuals through the within transform, only the idiosyncratic errors are left.

The default method for estimating σ_u^2 is

$$\widehat{\sigma}^2_{u\overline{T}} = \max\left\{0, \frac{\text{SSR}_b}{n-K} - \frac{\widehat{\sigma}_e^2}{\overline{T}}\right\}$$

where

$$\text{SSR}_b = \sum_{i=1}^{n}\left(\overline{y}_i - \widehat{\alpha}_b - \overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_b\right)^2$$

$\widehat{\alpha}_b$ and $\widehat{\boldsymbol{\beta}}_b$ are coefficient estimates from the between regression and $\overline{T}$ is the harmonic mean of T_i:

$$\overline{T} = \frac{n}{\sum_{i=1}^{n}\frac{1}{T_i}}$$

This estimator is consistent for σ_u^2 and is computationally less expensive than the second method. The sum of squared residuals from the between model estimate a function of both the idiosyncratic component and the individual component. Using our estimator of σ_e^2, we can remove the idiosyncratic component, leaving only the desired individual component.

The second method is the Swamy–Arora method for unbalanced panels derived by Baltagi and Chang (1994), which has a more precise small-sample adjustment. Using this method,

$$\widehat{\sigma}^2_{u\text{SA}} = \max\left\{0, \frac{\text{SSR}_b^* - (n-K)\widehat{\sigma}_e^2}{N - c_{\text{tr}}}\right\}$$

where

$$\text{SSR}_b^* = \sum_{i=1}^{n} T_i\left(\overline{y}_i - \widehat{\alpha}_b - \overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}_b\right)^2$$

$$c_{\text{tr}} = \text{trace}\left\{(\mathbf{X}'\mathbf{P}\mathbf{X})^{-1}\mathbf{X}'\mathbf{Z}\mathbf{Z}'\mathbf{X}\right\}$$

$$\mathbf{P} = \text{diag}\left\{\left(\frac{1}{T_i}\right)\boldsymbol{\iota}_{T_i}\boldsymbol{\iota}'_{T_i}\right\}$$

$$\mathbf{Z} = \text{diag}\left[\boldsymbol{\iota}_{T_i}\right]$$

$\mathbf{X}$ is the $N \times K$ matrix of covariates, including the constant, and $\boldsymbol{\iota}_{T_i}$ is a $T_i \times 1$ vector of ones.

The estimated coefficients $(\widehat{\alpha}, \widehat{\boldsymbol{\beta}})$ and their estimated covariance matrix $\widehat{\mathbf{V}}$ are reported together with the previously calculated quantities $\widehat{\sigma}_e$ and $\widehat{\sigma}_u$. The standard deviation of $\nu_i + e_{it}$ is calculated as $\sqrt{\widehat{\sigma}_e^2 + \widehat{\sigma}_u^2}$.

Reported as R^2 between is $\text{corr}(\overline{\mathbf{x}}_i\widehat{\boldsymbol{\beta}}, \overline{y}_i)^2$.

Reported as R^2 within is $\text{corr}\left\{(\mathbf{x}_{it} - \overline{\mathbf{x}}_i)\widehat{\boldsymbol{\beta}}, y_{it} - \overline{y}_i\right\}^2$.

Reported as R^2 overall is $\text{corr}(\mathbf{x}_{it}\widehat{\boldsymbol{\beta}}, y_{it})^2$.

xtreg, mle

The log likelihood for the ith unit is

$$l_i = -\frac{1}{2}\left(\frac{1}{\sigma_e^2}\left[\sum_{t=1}^{T_i}(y_{it} - \mathbf{x}_{it}\boldsymbol{\beta})^2 - \frac{{\sigma_u}^2}{T_i\sigma_u^2 + \sigma_e^2}\left\{\sum_{t=1}^{T_i}(y_{it} - \mathbf{x}_{it}\boldsymbol{\beta})\right\}^2\right] + \ln\left(T_i\frac{\sigma_u^2}{\sigma_e^2} + 1\right) + T_i\ln(2\pi\sigma_e^2)\right)$$

The `mle` and `re` options yield essentially the same results, except when total $N = \sum_i T_i$ is small (200 or less) and the data are unbalanced.

Similarly to `xtreg, fe` and `xtreg, re`, specifying `vce(robust)` or `vce(cluster` *clustvar*`)` causes the Huber/White/sandwich VCE estimator to be calculated for the estimated parameters in this regression.

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 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.

xtreg, pa

See [XT] **xtgee** for details on the methods and formulas used to calculate the population-averaged model using a generalized estimating equations approach.

Acknowledgments

We thank Richard Goldstein, who wrote the first draft of the routine that fits random-effects regressions, Badi Baltagi of the Department of Economics at Syracuse University, and Manuelita Ureta of the Department of Economics at Texas A&M University, who assisted us in working our way through the literature. We also thank Sergio Correia at the Federal Reserve Board for his work on controlling for high-dimensional categorical variables.

References

Alejo, J., A. F. Galvao, G. Montes-Rojas, and W. Sosa-Escudero. 2015. Tests for normality in linear panel-data models. *Stata Journal* 15: 822–832.

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

Andrews, M. J., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Arellano, M. 1987. Computing robust standard errors for within-groups estimators. *Oxford Bulletin of Economics and Statistics* 49: 431–434. https://doi.org/10.1111/j.1468-0084.1987.mp49004006.x.

———. 2003. *Panel Data Econometrics*. Oxford: Oxford University Press.

Baltagi, B. H. 1985. Pooling cross-sections with unequal time-series lengths. *Economics Letters* 18: 133–136. https://doi.org/10.1016/0165-1765(85)90167-3.

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

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

Baltagi, B. H., and Y.-J. Chang. 1994. Incomplete panels: A comparative study of alternative estimators for the unbalanced one-way error component regression model. *Journal of Econometrics* 62: 67–89. https://doi.org/10.1016/0304-4076(94)90017-5.

Baum, C. F. 2001. Residual diagnostics for cross-section time series regression models. *Stata Journal* 1: 101–104.

Bell, R. M., and D. F. McCaffrey. 2002. Bias reduction in standard errors for linear regression with multi-stage samples. *Survey Methodology* 28: 169–181.

Blackwell, J. L., III. 2005. Estimation and testing of fixed-effect panel-data systems. *Stata Journal* 5: 202–207.

Bottai, M., and N. Orsini. 2004. Confidence intervals for the variance component of random-effects linear models. *Stata Journal* 4: 429–435.

Bruno, G. S. F. 2005. Estimation and inference in dynamic unbalanced panel-data models with a small number of individuals. *Stata Journal* 5: 473–500.

Cabanillas, O. B., J. D. Michler, A. Michuda, and E. Tjernström. 2018. Fitting and interpreting correlated random-coefficient models using Stata. *Stata Journal* 18: 159–173.

Christodoulou, D., and V. Sarafidis. 2017. Regression clustering for panel-data models with fixed effects. *Stata Journal* 17: 314–329.

Correia, S. 2016. A feasible estimator for linear models with multi-way fixed effects. Unpublished manuscript, Duke University. https://scorreia.com/research/hdfe.pdf.

Correia, S., P. Guimarães, and T. Zylkin. 2020. Fast Poisson estimation with high-dimensional fixed effects. *Stata Journal* 20: 95–115.

De Hoyos, R. E., and V. Sarafidis. 2006. Testing for cross-sectional dependence in panel-data models. *Stata Journal* 6: 482–496.

Dwyer, J. H., and M. Feinleib. 1992. Introduction to statistical models for longitudinal observation. In *Statistical Models for Longitudinal Studies of Health*, ed. J. H. Dwyer, M. Feinleib, P. Lippert, and H. Hoffmeister, 3–48. New York: Oxford University Press.

Gardner, J. 2023. Stata tip 149: Weighted estimation of fixed-effects and first-differences models. *Stata Journal* 23: 276–280.

Hansen, B. E. 2022. *Econometrics*. Princeton, NJ: Princeton University Press.

Hernández-Ramos, L. M., R. Escalante, and M. Raydan. 2011. Unconstrained optimization techniques for the acceleration of alternating projection methods. *Numerical Functional Analysis and Optimization* 32: 1041–1066. https://doi.org/10.1080/01630563.2011.591954.

Hoechle, D. 2007. Robust standard errors for panel regressions with cross-sectional dependence. *Stata Journal* 7: 281–312.

Hughes, R. A., M. G. Kenward, J. A. C. Sterne, and K. Tilling. 2017. Analyzing repeated measurements while accounting for derivative tracking, varying within-subject variance, and autocorrelation: The xtmixediou command. *Stata Journal* 17: 573–599.

Judge, G. G., W. E. Griffiths, R. C. Hill, H. Lütkepohl, and T.-C. Lee. 1985. *The Theory and Practice of Econometrics*. 2nd ed. New York: Wiley.

Lee, L.-F., and W. E. Griffiths. 1979. The prior likelihood and best linear unbiased prediction in stochastic coefficient linear models. Working paper 1, Department of Econometrics, Armidale, Australia: University of New England.

Libois, F., and V. Verardi. 2013. Semiparametric fixed-effects estimator. *Stata Journal* 13: 329–336.

Magazzini, L., R. L. Bruno, and M. Stampini. 2020. Using information from singletons in fixed-effects estimation: xtfesing. *Stata Journal* 20: 965–975.

McCaffrey, D. F., K. Mihaly, J. R. Lockwood, and T. R. Sass. 2012. A review of Stata commands for fixed-effects estimation in normal linear models. *Stata Journal* 12: 406–432.

Nichols, A. 2007. Causal inference with observational data. *Stata Journal* 7: 507–541.

Pinzon, E. 2015. Fixed effects or random effects: The Mundlak approach. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2015/10/29/fixed-effects-or-random-effects-the-mundlak-approach/.

Rios-Avila, F. 2015. Feasible fitting of linear models with N fixed effects. *Stata Journal* 15: 881–898.

Schunck, R. 2013. Within and between estimates in random-effects models: Advantages and drawbacks of correlated random effects and hybrid models. *Stata Journal* 13: 65–76.

Stammann, A. 2018. Fast and feasible estimation of generalized linear models with high-dimensional k-way fixed effects. Unpublished manuscript. https://arxiv.org/pdf/1707.01815.pdf.

Stock, J. H., and M. W. Watson. 2008. Heteroskedasticity-robust standard errors for fixed effects panel data regression. *Econometrica* 76: 155–174. https://doi.org/10.1111/j.0012-9682.2008.00821.x.

Swamy, P. A. V. B., and S. S. Arora. 1972. The exact finite sample properties of the estimators of coefficients in the error components regression models. *Econometrica* 40: 261–275. https://doi.org/10.2307/1909405.

Taub, A. J. 1979. Prediction in the context of the variance-components model. *Journal of Econometrics* 10: 103–107. https://doi.org/10.1016/0304-4076(79)90068-X.

Twisk, J. W. R. 2013. *Applied Longitudinal Data Analysis for Epidemiology: A Practical Guide*. 2nd ed. Cambridge: Cambridge University Press.

Wooldridge, J. M. 2020. *Introductory Econometrics: A Modern Approach*. 7th ed. Boston: Cengage.

Wursten, J. 2018. Testing for serial correlation in fixed-effects panel models. *Stata Journal* 18: 76–100.

Also see

[XT] **xtreg postestimation** — Postestimation tools for xtreg+

[XT] **xteregress** — Extended random-effects linear regression

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

[XT] **xtgls** — GLS linear model with heteroskedastic and correlated errors

[XT] **xtheckman** — Random-effects regression with sample selection

[XT] **xtivreg** — Instrumental variables and two-stage least squares for panel-data models

[XT] **xtregar** — Fixed- and random-effects linear models with an AR(1) disturbance

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

[BAYES] **bayes: xtreg** — Bayesian random-effects linear model

[ME] **mixed** — Multilevel mixed-effects linear regression

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

[R] **areg** — Linear regression with many indicator variables+

[R] **regress** — Linear regression

[R] **wildbootstrap** — Wild cluster bootstrap inference

[SP] **spxtregress** — Spatial autoregressive models for panel data

[TS] **forecast** — Econometric model forecasting

[TS] **prais** — Prais–Winsten and Cochrane–Orcutt regression

[U] **20 Estimation and postestimation commands**

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