xtdpdml: Linear Dynamic Panel-Data Estimation using Maximum Likelihood and Structural Equation Modeling

Richard Williams, University of Notre Dame (rwilliam@nd.edu)
Paul D. Allison, University of Pennsylvania (allison@statisticalhorizons.com)
Enrique Moral-Benito, Banco de Espana, Madrid (enrique.moral@gmail.com)
Last revised November 21, 2016

Abstract

Panel data make it possible both to control for unobserved confounders and to include lagged, endogenous regressors. Trying to do both at the same time, however, leads to serious estimation difficulties. In the econometric literature, these problems have been addressed by using lagged instrumental variables together with the generalized method of moments (GMM), while in sociology the same problems have been dealt with via maximum likelihood estimation and structural equation modeling. While both approaches have merit, we show that the ML (SEM) method is substantially more efficient than the GMM method when the normality assumption is met, and it also suffers less from finite sample biases. We introduce a command named `xtdpdml` with syntax similar to other Stata commands for linear dynamic panel-data estimation. `xtdpdml` greatly simplifies the SEM model specification process; makes it possible to test and relax many of the constraints that are typically embodied in dynamic panel models; allows for the inclusion of time-invariant variables in the model, unlike most related methods; and takes advantage of Stata's ability to use full information maximum likelihood for dealing with missing data. The strengths and advantages of `xtdpdml` are illustrated via examples from both economics and sociology.

1. Introduction

Panel data make it possible both to control for unobserved confounders and to include lagged, endogenous regressors. Trying to do both at the same time, however, leads to serious estimation difficulties. In the econometric literature, these problems have been addressed by using lagged instrumental variables together with the generalized method of moments (GMM). In Stata, commands such as `xtabond, xtdpdsys` and `xtdpd` have been used for these models.

Perhaps reflecting historical disciplinary differences, sociologists (Allison, 2009; Bollen and Brand, 2010) have often taken a different approach. As Allison (2009; in progress) has shown, the same problems can be dealt with via maximum likelihood estimation and structural equation modeling (SEM). The ML (SEM) method is substantially more efficient than the GMM method when the normality assumption is met and suffers less from finite sample biases. In Stata, the `sem` command can be used for this purpose. Unfortunately, the process for specifying these models with `sem` is extremely tedious and error prone.

In this paper we introduce a new command, `xtdpdml`, which fits dynamic panel data models using maximum likelihood. It works as a shell for `sem`, generating the necessary commands. It

can also generate code for running these models in Mplus—a popular stand-alone package for structural equation modeling. `xtdpdml` tends to work best when panels are strongly balanced, the number of time points is relatively small (e.g. less than 10), and there are no missing data. But it can also often work well when these conditions are not met. The multidisciplinary strengths and advantages of `xtdpdml` are illustrated via examples from both economics and sociology.

`xtdpdml` greatly simplifies the SEM model specification process; makes it possible to test and relax many of the constraints that are typically embodied in dynamic panel models; allows for the inclusion of time-invariant variables in the model, unlike most related fixed effects methods; and takes advantage of Stata's ability to use full information maximum likelihood (FIML) for dealing with missing data. `xtdpdml` can also estimate models involving reciprocal causation and is sometimes superior to the `xtreg` command when data are missing or when time-invariant variables are employed. `xtdpdml` also provides an overall goodness of fit measure by default and provides access to others via the `sem` postestimation command `estat gof, stats(all)`. Many other `sem` postestimation commands can be used as well.

2. The xtdpdml model

Panel data have two major attractions for making causal inferences: the ability to control for unobserved, time-invariant confounders, and the ability to estimate models with lagged, endogenous regressors—which can be helpful in making inferences about causal direction.

Controlling for unobservables can be accomplished with well-known fixed effects methods (such as the linear fixed effects model that can be optionally estimated with `xtreg`). For examining causal direction, the most popular approach has long been the cross-lagged panel model. In cross-lagged panel models, *x* and *y* at time *t* affect both *x* and *y* at time *t+1*. Economists typically refer to such models as dynamic panel models because of the lagged effect of the dependent variable on itself.

Unfortunately, attempting to combine fixed effects models with cross-lagged panel models leads to serious estimation problems. The estimation difficulties include error terms that are correlated with predictors, the so-called “incidental parameters problem”, and uncertainties about the treatment of initial conditions (Allison, in progress; also see Wooldridge (2010), Baltagi (2013), or Hsiao (2014) for additional review of the extensive literature on dynamic panel data models).

The most popular econometric method for estimating dynamic panel models is the generalized method of moments (GMM) that relies on lagged variables as instruments. This method has been incorporated into several commercial software packages, usually under the name of Arellano-Bond (A-B) estimators. For example, Stata has the built-in `xtabond` command and the user-written `xtabond2` command.

While the A-B approach provides consistent estimators of the coefficients, there is substantial evidence that the estimators are not fully efficient (Ahn and Schmidt 1995) and often perform poorly when the autoregressive parameter (the effect of a variable on itself at a later point in time) is near 1.0.

Moral-Benito (2013; Moral-Benito et al., in progress; also see Bai 2013) shows that maximum likelihood estimation can be accomplished in a way that eliminates the incidental parameters problem without the need for special assumptions about initial conditions. Moral-Benito uses two equations to write his model[1]. They are

(1) $$y_{it} = \lambda y_{it-1} + x'_{it}\beta + \omega_i'\delta + \alpha_i + \xi_t + \upsilon_{it}$$

where

y_{it-1} is the value of *y* for individual *i* at time *t*-1

x_{it} is a vector of sequentially exogenous/predetermined time-varying variables

ω_i is a vector of time-invariant strictly exogenous variables

α_i is the unobservable time-invariant fixed effect

ξ_t captures unobserved common factors across units in the panel

υ_{it} is the time-varying error term

(2) $$E(\upsilon_{it} \mid y_i^{t-1}, x_i^{t}, \omega_i, \alpha_i) = 0 \ \left(t = 1,...,T\right)\left(i = 1,...,N\right)$$

where x_i^t denotes a vector of the observations accumulated up to *t*. This implies, for example, that the residual for y_5 is uncorrelated with predetermined variable *x* at times 1-5, but could be correlated with *x* at later times, e.g. x_6, x_7, etc. Put another way, the predetermined variable *x* could be affected by earlier values of the dependent variable. The meaning of each type of variable will become clearer as we move along. Further details on the econometric specification and the resulting likelihood function are provided by Moral-Benito et al. (in progress).

Condition (2) is the only assumption required for consistency and asymptotic normality (under fixed *T* when *N* tends to infinity). Moral-Benito's (2013) model does NOT include strictly exogenous time-varying variables but it can be easily modified to do so.

Allison (in progress) further shows that the dynamic panel model is a special case of the general linear structural equation model (SEM) and that the method of Moral-Benito can be implemented (and extended) with Stata's `sem` command. Allison (in progress) and Moral-Benito (2013) claim that the SEM approach has several advantages over both GMM methods and previous ML methods: there is no "incidental parameters" problem; initial conditions are treated as completely exogenous and do not need to be modeled; no difficulties arise when the autoregressive parameter (the effect of lagged *y* on *y*) is at or near 1.0; missing data are easily handled by full-information maximum likelihood; coefficients can be estimated for time-invariant predictors (the standard A-B method cannot do this because it uses difference scores which causes all time-invariant variables to drop out); and many model constraints can be easily relaxed and/or tested.

[1] Here and elsewhere we have slightly modified Moral-Benito's notation to make it consistent with `xtdpdml`'s.

Further, it is well known that likelihood-based approaches (ML) are preferred to method-of-moments (GMM) counterparts in terms of finite-sample performance (see Anderson, Kunitomo, and Sawa 1982), and that ML is more efficient than GMM under normality. Moral-Benito (2013) compares the widely-used panel GMM estimator of Arellano-Bond (1991) with its likelihood-based counterpart and confirms these results in the case of dynamic panel models with predetermined regressors.

To show specifically how the SEM approach can be used in Stata, Allison (in progress) reanalyzes data described by Cornwell and Rupert (1988) for 595 household heads who reported a non-zero wage in each of 7 years from 1976 to 1982. The variables are wks = number of weeks employed in each year; union = 1 if wage set by union contract, else 0, in each year; lwage = ln(wage) in each year; and ed = years of education in 1976.

Here is the Stata `sem` code (Adapted from Allison, in progress):

```
use http://www3.nd.edu/~rwilliam/statafiles/wages, clear
keep wks lwage union ed id t
reshape wide wks lwage union, i(id) j(t)
sem    (wks2 <- wks1@b1 lwage1@b2 union1@b3 ed@b4 Alpha@1 E2@1) ///
       (wks3 <- wks2@b1 lwage2@b2 union2@b3 ed@b4 Alpha@1 E3@1) ///
       (wks4 <- wks3@b1 lwage3@b2 union3@b3 ed@b4 Alpha@1 E4@1) ///
       (wks5 <- wks4@b1 lwage4@b2 union4@b3 ed@b4 Alpha@1 E5@1) ///
       (wks6 <- wks5@b1 lwage5@b2 union5@b3 ed@b4 Alpha@1 E6@1) ///
       (wks7 <- wks6@b1 lwage6@b2 union6@b3 ed@b4 Alpha@1), ///
       var(e.wks2@0 e.wks3@0 e.wks4@0 e.wks5@0 e.wks6@0) var(Alpha) ///
       cov(Alpha*(ed)@0) cov(Alpha*(E2 E3 E4 E5 E6)@0) ///
       cov(_OEx*(E2 E3 E4 E5 E6)@0) cov(E2*(E3 E4 E5 E6)@0) ///
       cov(E3*(E4 E5 E6)@0) cov(E4*(E5 E6)@0) cov(E5*(E6)@0) ///
       cov(union3*(E2)) cov(union4*(E2 E3)) cov(union5*(E2 E3 E4)) ///
       cov(union6*(E2 E3 E4 E5)) ///
       iterate(250) technique(nr 25 bhhh 25) noxconditional
```

We will explain the different components of the model in a moment, but even just glancing at the code underscores the difficulty of the task. For the SEM approach, data need to be in wide format; many/most dynamic panel data sets will be in long format. Coding is lengthy and error prone; there is a separate equation for each time period, there are many constraints across equations, and getting the covariance structure right is especially difficult. Output (not shown) is voluminous and highly repetitive because of the many equality constraints across time. Limitations of Stata make the coding less straightforward than we might like. Stata won't allow covariances between predetermined *x*s (to be defined shortly) and the *y* residuals. The `xtdpdml` command therefore zeroes out most of the *y* residuals and replaces them with latent exogenous variables (E2, E3, etc.).

`xtdpdml` avoids most of these problems. Here is equivalent coding using `xtdpdml` and the resulting output:

```
. use http://www3.nd.edu/~rwilliam/statafiles/wages, clear
. xtset id t
       panel variable:  id (strongly balanced)
        time variable:  t, 1 to 7
                delta:  1 unit
```

```
. xtdpdml wks L.lwage, inv(ed) pre(L.union)

Highlights: Dynamic Panel Data Model using ML for outcome variable wks
------------------------------------------------------------------------------
             |                 OIM
         wks |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
wks          |
         wks |
          L1. |   .1871266   .0201939     9.27   0.000     .1475473    .2267059
             |
       lwage |
          L1. |   .6417917   .4842304     1.33   0.185    -.3072823    1.590866
             |
       union |
          L1. |  -1.191349   .5168951    -2.30   0.021    -2.204445   -.1782536
             |
           ed |  -.1122267   .0559477    -2.01   0.045    -.2218822   -.0025711
------------------------------------------------------------------------------
# of units = 595. # of periods = 7. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(71)  =     110.23, Prob > chi2 =  0.0020
IC Measures: BIC =   25470.43  AIC =   24772.64
Wald test of all coeff = 0: chi2(4) =      90.09, Prob > chi2 =  0.0000
```

One short command generates the equivalent of the 13 lines of sem code shown earlier. `xtdpdml` also temporarily reshaped the data to wide format.

Unless the user requests otherwise, only the most critical output is shown. By default, all variable coefficients (but not the constants or the error variances) are constrained to be equal across time. Therefore only the first equation (in this case for time 2) needs to be presented. The LR statistic provides an overall goodness of fit test. This tests all the constraints on the variances and covariances that are implied by the model. The BIC and AIC statistics (which could also be obtained via the `estat ic` command) are included in the output. (Note that these statistics could not be computed correctly if you were using a highlights-only file, described shortly). The Wald statistic tests the null hypothesis that all the variables in the model have coefficients of zero. In this case, where coefficients are constrained to be the same across all time periods, it produces the same results as the `sem` post-estimation command `estat eqtest`. When some coefficients are free to differ across time periods `estat eqtest` provides a test for each time period separately whereas `xtdpdml` tests all coefficients for all times simultaneously.

`xtdpdml` obviously provides a much simpler syntax. The reason it isn't simpler still (and why the `sem` coding is so difficult) is that there are several possible types of independent variables in the model:

The lag 1 value of *y* (e.g. L1.wks) is included by default. This can be changed with the `ylag` option, e.g. `ylag(1 2)`, `ylag(2 4)`. Specifying `ylag(0)` excludes all lagged values of *y*.

Strictly exogenous variables are those that (by assumption) are uncorrelated with the error terms at all points in time. Equivalently, we assume that they are not affected by prior values of the dependent variable. These variables are listed immediately after the dependent variable, before

the comma. Time series notation can be used, e.g. `xtdpdml y L1.lwage L2.lwage` would include the first and second lagged values of wages as independent variables.

Predetermined variables, also known as sequentially or weakly exogenous, are variables that can be affected by prior values of the dependent variables. In the current example, we allow for the possibility that weeks worked in one year can affect union status in later years. Predetermined variables are specified with the `pre` option. Mechanically, the *y* residuals are allowed to correlate with the later-in-time values of the predetermined variables.

Time-invariant variables are variables whose values are constant across time, such as year born. In the current example, years of education does not vary across time. These are specified with the `inv` option. The ability to use time-invariant variables in the model is one of the advantages of the SEM approach over methods based on first differences like Arellano-Bond.

Also automatically included in each model is the latent exogenous variable Alpha. Alpha represents the "fixed effects" that are common to all equations across time. Alpha can freely covary with all the time-varying observed exogeneous variables (both strictly exogenous and predetermined), but not with the time-invariant observed exogeneous variables. As Allison (in progress) says, "This is exactly what we want to achieve in order for Alpha to truly behave as a set of fixed effects". To further clarify, Allison (2009, pp. 2-3) explains that

> In a random effects model, the unobserved variables are assumed to be uncorrelated with (or, more strongly, statistically independent of) all the observed variables. In a fixed effects model, the unobserved variables are allowed to have any association whatever with the observed variables (which turns out to be equivalent to treating the unobserved variables as fixed parameters.) Unless you have controlled for such associations, you haven't really controlled for the effects of the unobserved variables. This is what makes the fixed effects approach so attractive.

3. The xtdpdml command and syntax

The general syntax is

```
   xtdpdml y [time-varying strictly exogeneous vars] [,
inv(time-invariant exogenous vars) pre(predetermined vars)
other_options]
```

Following is a description of the numerous program options.

Independent variables (other than strictly exogenous)

`inv(`*`varlist`*`)` Time-invariant exogenous variables, e.g. year of birth.

`predet(`*`varlist`*`)` Predetermined variables, also known as sequentially exogenous. Predermined variables can be affected by prior values of the dependent variable. Time series notation can be used.

`ylag(`*`numlist`*`)` By default the lag 1 value of *y* is included as an independent variable. Different or multiple lags can be specified, e.g. `ylag(1 2)` would include lags 1 and 2 of *y*. `ylag(0)` will cause no lagged value of *y* to be included in the model.

Dataset Options

`wide` By default, data are assumed to be xtset long with both time and panelid variables specified. The data set is temporarily converted to wide format for use with sem. If data are already in wide format use the `wide` option. However, note that the file must have been created by a `reshape wide` command or else it won't have information that `xtdpdml` needs. Use of this option is generally discouraged.

`staywide` will keep the data in wide format after running `xtdpdml`. This may be necessary if you want to use post-estimation commands like `predict`.

`tfix` Time should be coded t = 1, 2, ..., T where T = number of waves. By default, units like years (e.g. 1990, 1991, 1992) will cause errors or incorrect results. There will also be errors or incorrect results if delta does not equal 1, e.g. t = 1, 3, 5. The `tfix` option will recode time to equal 1, 2, ..., T and set delta = 1. You can still have problems though if delta was not specified correctly in the source data set or if interval width is not consistent. It is safest if you correctly code time yourself but `tfix` should work in most cases.

`std` standardizes all the variables in the model to have mean 0 and variance 1. It does this while the data set is in long format, hence the standardization does NOT differ by time period; e.g. at all time periods you might subtract 10 from a variable and divide by 7. By standardizing this way, the coefficients remain comparable across time. You probably will not want to use this option in most cases, but it can sometimes help when the model is having trouble converging. Does not work if the `wide` option has been specified, i.e., data are already in wide format.

`std(`*`varlist`*`)` standardizes only the selected variables to have mean 0 and variance 1. Does not work if the `wide` option has been specified. Do NOT use time series notation; just list the names of the variables you want standardized.

Model Specification and Constraints Options

`evars` sometimes helps with convergence when there are no predetermined variables in the model. It is an alternative and usually less efficient way of specifying the error terms. But sometimes it helps and may be necessary for replicating results from earlier versions of the program.

`alphafree` lets the coefficients of Alpha (fixed effects) differ across time. Note that, if this option is used, Alpha will be normalized by fixing its variance at 1.

`xfree` lets the coefficients of all the independent variables (except lagged *y*) freely differ across time.

`xfree(`*`varlist`*`)` lets the coefficients of the specified independent variables freely differ across time.

`yfree` lets all lagged y coefficients freely differ across time.

`yfree(`*`numlist`*`)` allows the specified lagged y coefficients to freely differ across time.

`nocsd` (alias is `constinv`) Cross-sectional dependence is NOT allowed, i.e., constants are constrained to be equal across waves. This is equivalent to no effect of time. This option sometimes causes convergence problems.

`errorinv` constrains error variances to be equal across waves. The default is to let them freely differ. This option may cause convergence problems.

`re` estimates the Random Effects Model (where Alpha is uncorrelated with all observed Xs)

Reporting Options

`title(`*`string`*`)` Gives a title to the analysis. This title will appear in both the highlights results and (if requested) the Mplus code. For example, `ti(Baseline Model)`

`details` will show all the output generated by the `sem` command. Otherwise only a highlights version is presented. This can be useful if you want to make sure the model specification is correct or if you want information not contained in the highlights. You can also replay all the results just by typing `sem` after running `xtdpdml`.

`showcmd` will show the sem command generated by `xtdpdml`. This can be useful to make sure the estimated model is what you wanted.

`gof` reports several goodness of fit measures after model estimation. It has the same effect as running the `sem` postestimation command `estat gof, stats(all)` after `xtdpdml`.

`tsoff` By default, when possible the highlights output produced by `xtdpdml` will use time-series notation similar to what you see with commands like `xtabond`, e.g. L3.xvar will represent the lag 3 value of xvar. Since the data are reshaped wide, this is not the same as the name of the variable that was actually used, e.g. it might be that L3.xvar corresponds to xvar2. `tsoff` will turn off the use of time series notation in the highlights printout and show the names of the variables actually used in the reshaped wide data.

display_options include noci, nopvalues, noomitted, vsquish, noemptycells, cformat(%fmt), pformat(%fmt), sformat(%fmt), and nolstretch; see [R] estimation options.

coeflegend displays the names of the coefficients instead of the inferential statistics. This can be useful if, say, you are trying to use post-estimation test commands to test hypotheses about effects.

decimals(*integer*) specifies the number of decimal places to display for the coefficients, standard errors, and confidence limits. It is a shorthand way of specifying cformat, e.g. dec(*3*) is the same as specifying cformat(*%9.3f*). You will get an error if you specify both dec and cformat. The value specified must range between 0 and 8; 3 is often a good choice for making the output easier to read.

Other Options

mplus(*filenamestub, mplus options*) will create inp and data files that can be used by Mplus (has only been tested with Mplus 7.4). This is adapted (with permission) from UCLA's and Michael Mitchell's stata2mplus command but does not require that it be installed. The filenamestub must be specified; it will be used to name the Mplus .inp and .dat files. Everything else is optional. Options replace, missing(#), listwise, analysis, and output are supported. replace will cause existing .inp and .dat files to be overwritten. missing specifies the missing value for all variables; default is -9999. listwise will cause listwise deletion to be used rather than fiml. analysis and output specify options to be passed to the Mplus analysis and output options. As is the case in Mplus, multiple analysis and output options should be separated by semicolons. xtdpdml cannot check your Mplus syntax so be careful. The Mplus option, of course, requires that you have Mplus and know how to use it. Since that will not be true of many/most Stata users, those interested in the option should consult the help file and examples provided on the support page for xtdpdml for additional details.

semfile(filename, r) The generated sem commands will be output to a file called filename.do. The r option can be specified to replace an existing do file by that name. This is useful if you want to try to modify the sem commands in ways that are not easily done with xtdpdml. You may wish to also specify the staywide option so that data remain correctly formatted for use with the generated do file.

store(*stubname*) - xtdpdml generates two sets of results: the full results, generated by sem, and a highlights-only set of results which can be used with programs like esttab. The stored results have the names stubname_f and stubname_h, e.g. if you specify store(*model1*) the results will be stored as model1_f and model1_h. The default stubname is xtdpdml, so after running xtdpdml without the store option you

should have stored results xtdpdml_f and xtdpdml_h. You shouldn't try to do any post-estimation commands with the highlights version (e.g. `predict`, `margins`) because necessary information may not be stored in the file; use the full version instead.

`dryrun` will keep `sem` from actually being executed. This will catch some errors immediately and can be useful if you want to see the `sem` command that is generated and/or wish to specify `staywide` to reformat the data from long to wide. This will often be combined with the `showcmd`, `mplus`, `semfile`, or `staywide` options.

`iterate(#)` specifies the maximum number of iterations allowed. The current default (subject to change) is 250. You can increase this number and/or change the maximization technique if the model is having trouble converging.

`technique(`*`methods`*`)` specifies the maximization techniques used. The current default (subject to change) is `technique(nr 25 bhhh 25)`. You can change this if the model is having trouble converging. See `help maximize` for details as well as for information on other options that can be used, e.g. `difficult`.

`semopts(`*`options`*`)` Other options allowed by `sem` will be included in the generated `sem` command.

`fiml` causes Full Information Maximum Likelihood to be used for missing data. This is the equivalent of specifying `method(mlmv)` on the `sem` command. `fiml` sometimes dramatically slows down execution so be patient if you use it.

`skipcfatransform` and `skipconditional` – Stata 14.2 changed the way starting values are computed by `sem`. When used together, `skipcfatransform` and `skipconditional` cause Stata to compute starting values the same way as it did before Stata 14.2. Usually the new procedures work better, especially when fiml is used, but sometimes the old start values speed up execution and/or are better for getting models to converge. These options are ignored in Stata 14.1 or earlier.

`altstart` is a convenient way to specify both `skipcfatransform` and `skipconditional`.

`method(`*`method`*`)`specifies estimation methods supported by sem, e.g. ml, mlmv, adf. You probably will not use this option unless you want to specify `method(adf)`. Remember that `method(ml)` (maximum likelihood) is the default and that `fiml` is a shorthand way of specifying `method(mlmv)` (maximum likelihood with missing values, aka full information maximum likelihood). If you use `method(adf)` (asymptotic distribution free) you may also have to use the `technique` option since adf and the bhhh technique do not seem to work together.

`vce(vcetype)` specifies vcetypes supported by sem, e.g. oim, robust. Not all vcetypes have been tested with `xtdpdml` so we recommend caution if using this option.

`v12` The `xtdpdml` command was written and tested using Stata 13 and 14. The `v12` option will also allow it to run under Stata 12.1. This has not been extensively tested so use at your own risk.

4. Examples

We have already provided one example that illustrates the key features of `xtdpdml`. For many purposes, that one example may be enough. Here, we illustrate additional capabilities of `xtdpdml` that will often be useful.

4.1 Comparisons with xtabond – Simulated Data.

Drawing on work from Bun and Kiviet (2006) and Moral-Benito (2013), Moral-Benito et al. (in progress) offer several simulations comparing ML estimates (using `xtdpdml`) with GMM estimates (using `xtdpd`—the built-in Stata command). Here we present a modified replication of part of their analyses. The data for the dependent variable *y* and the independent variable *x* are generated according to the equations

$$y_{it} = \lambda y_{it-1} + \beta x_{it} + \alpha_i + \upsilon_{it}$$
$$x_{it} = \rho x_{it-1} + \phi y_{it-1} + \pi \alpha_i + \xi_{it}$$

where υ_{it}, ξ_{it}, and α_i are generated as $\upsilon_{it} \sim i.i.d(0,1)$, $\xi_{it} \sim i.i.d.(0, 6.58)$, and $\alpha_i \sim (0, 2.96)$. This configuration allows for fixed effects correlated with the regressor as well as feedback from *y* to *x*. The parameter λ reflects the effect of the lagged dependent variable on the dependent variable (i.e. the autoregressive effect), β reflects the effect of the independent variable on the dependent variable, ρ is the effect of lagged *x* on current *x*, ϕ captures the feedback from the lagged dependent variable to the regressor, and π reflects the association between the fixed effects and the regressor.

In the following, we fix $\lambda = 0.75$, $\beta = 0.25$, $\rho = 0.5$, $\phi = -0.17$, and $\pi = 0.67$. We do 5000 simulations where the value of *N* varies from as low as 100 to as much as 5,000. The complete code for the simulation program, as well as the commands used to execute it, are in the appendix. The key estimation commands included in the code are

```
capture xtdpdml y, pre(x) skipcfatransform skipconditional constinv
xtabond y, pre(x)
```

The `xtdpdml` command includes the options `skipcfatransform` and `skipconditional`, which, in Stata 14.2, change the way start values are computed. Usually these options are not necessary but we found that in this case they helped with convergence. We used the `constinv` option (constants the same across time) to make the comparisons between `xtdpdml` and `xtabond` as fair as possible. Alternatively, as explained in section 5.1, we could have added time dummies to `xtabond`. Using the `xtdpd` command instead of `xtabond` produced identical results; we used `xtabond` because its syntax is more similar to `xtdpdml`'s.

Here are the key results:

```
N = 100, T = 4, bylag = .75, bx = .25, 5000 simulations

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     biasyml |      4,121   -.0057041    .1642201    -.44497   .6834952
    biasygmm |      5,000   -.2192548    .2899418  -1.545341   1.423459
     biasxml |      4,121    .0029312    .0917246  -.3246409   .3360264
    biasxgmm |      5,000   -.0841894    .1290918  -.6902781   .5682601

Root Mean Squared Error of lagged y ml  = .16431918
Root Mean Squared Error of lagged y gmm = .36350915
Root Mean Squared Error of x ml         = .09177143
Root Mean Squared Error of x gmm        = .15411863
```

In the moderate *N* / small *T* case, the ML estimators produced by `xtdpdml` show virtually no bias. The average ML estimate of the autoregressive effect λ (true value = .75) is off by less than .01, while the GMM estimate is off by an average of almost .22. For the effect of *x* on *y*, β (true value is .25), the average ML estimate is off by less than .003 while the average GMM estimate is off by more than .08. The standard deviations (and also the inter-quartile ranges, not shown) are also smaller for the ML estimates than the GMM estimates. The root mean squared errors (which are affected by both bias and variability in estimates across sample) indicate that, across simulations, the errors in estimating the coefficients are about twice as large for GMM as they are for ML.

However, it should also be noted that in almost 20 percent of the simulations, the ML estimates failed to converge to a solution. As we explain later, there are approaches that may achieve convergence when the default `xtdpdml` options fail to do so. This would, however, be quite tedious to try with nearly nine hundred different simulations that did not converge. In order to make sure that the non-converging simulations were not giving the ML estimates an unfair advantage (e.g., maybe non-converging simulations were ones where the `xtdpdml` estimates were most biased), we also ran analyses where we excluded the non-converging `xtdpdml` simulations in the bias estimates for GMM. ML actually fared slightly better relative to GMM when this was done.

We also ran several other simulations. First, we increased the sample size to 500:

```
N = 500, T = 4, bylag = .75, bx = .25, 5000 simulations

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     biasyml |      4,553    .0105436    .1029967  -.2903285     .37156
    biasygmm |      5,000   -.0733646    .1742834  -.6856158    .737546
     biasxml |      4,553    .0054282    .0492591  -.1340084   .1911209
    biasxgmm |      5,000   -.0289765    .0758509  -.3016998   .3145986

Root Mean Squared Error of lagged y ml  = .10353497
Root Mean Squared Error of lagged y gmm = .18909539
Root Mean Squared Error of x ml         = .04955724
Root Mean Squared Error of x gmm        = .08119725
```

The GMM biases are much smaller than before, but the GMM estimates were still more biased and had larger standard deviations and RMSEs than the ML estimates. Convergence was much less of a problem for ML with the larger sample size.

We then increased the sample size to 1,000:

```
N = 1000, T = 4, bylag = .75, bx = .25, 5000 simulations

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     biasyml |      4,749    .0161765    .0899964  -.1972505   .2767218
    biasygmm |      5,000   -.0387785    .1321643  -.4553157   .6357055
     biasxml |      4,749    .0071507    .0409448  -.1010884   .1462057
    biasxgmm |      5,000    -.015212    .0573026  -.2033475   .2983379

Root Mean Squared Error of lagged y ml  = .09143863
Root Mean Squared Error of lagged y gmm = .13773586
Root Mean Squared Error of x ml         = .04156457
Root Mean Squared Error of x gmm        = .05928741
```

The GMM bias continued to decline, but ML still produced slighter better estimates on average, with the ML biases being only about half as large as the GMM biases. The larger sample size continued to make it more likely that ML simulations would converge.

Next, we increased the sample size to 5,000:

```
N = 5000, T = 4, bylag = .75, bx = .25, 5000 simulations

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     biasyml |      4,996     .010794    .0553024  -.1203409   .2192318
    biasygmm |      5,000   -.0079405    .0616096  -.2535154   .2166942
     biasxml |      4,996    .0045946    .0239609  -.0555138   .0996841
    biasxgmm |      5,000   -.0030869    .0263759  -.1009593   .0891423

Root Mean Squared Error of lagged y ml  = .05634593
Root Mean Squared Error of lagged y gmm = .06211917
Root Mean Squared Error of x ml         = .02439748
Root Mean Squared Error of x gmm        = .02655593
```

In this case, the differences in bias between ML and GMM were trivial. Also almost all of the ML simulations converged.

We also ran simulations where we increased *T*. We found that as *T* increased, the biases of the GMM method declined. For example,

```
N = 100, T = 10, bylag = .75, bx = .25, 5000 simulations

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     biasyml |      4,942    .0080347    .0529419  -.1304359   .2153849
    biasygmm |      5,000   -.0507636      .04587  -.2283615    .092642
     biasxml |      4,942    .0019953    .0217144  -.0832179   .0897856
    biasxgmm |      5,000   -.0077148    .0189994  -.0828093    .056326

Root Mean Squared Error of lagged y ml  = .05354811
Root Mean Squared Error of lagged y gmm = .06841784
Root Mean Squared Error of x ml         = .02180586
Root Mean Squared Error of x gmm        = .02050596
```

With $T = 10$ and $N = 100$, the ML estimates were still less biased than the GMM estimates, but the ML advantage was much smaller than it was with $T = 4$. Other simulations showed that as T further increases the advantages of ML over GMM continue to get smaller.

Similarly, we found that, as the autoregressive effect got larger, GMM estimates became more biased; but conversely, GMM estimates were less biased and the advantages of ML were smaller (or non-existent) when the autoregressive effect was smaller. For example, when the autogressive effect was only .50,

```
N = 100, T = 4, bylag = .5, bx = .25, 5000 simulations

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
     biasyml |      4,459    .0692521    .1840769  -.3794991   .8573828
    biasygmm |      5,000   -.1066614    .2149693  -.9683192   1.048151
     biasxml |      4,459    .0340245    .0966074  -.2971291   .3823025
    biasxgmm |      5,000   -.0388968    .1009658  -.5280045   .6149718

Root Mean Squared Error of lagged y ml  = .19667274
Root Mean Squared Error of lagged y gmm = .23997592
Root Mean Squared Error of x ml         = .10242394
Root Mean Squared Error of x gmm        = .10819914
```

In short, with moderate N and small T, the ML approach produces smaller biases in estimates than does GMM. As N and/or T gets larger, or as the autoregressive effect gets smaller, the advantages of ML over GMM diminish. However, our next example suggests that, even with large N, there may be situations in which the ML approach should be preferred.

4.2 Comparisons with xtabond – Real Data. Of course, real data often offer complications that are not present in simulations. The following example is adapted from Bollen and Brand (2010). They examine data from the National Longitudinal Survey of Youth. Respondents were 14 to 22 years old when first interviewed in 1979, and were interviewed annually or bi-annually for several years thereafter. Bollen and Brand originally analyzed data from the years 1983, 1985, 1987, 1989, 1991 and 1993. The dependent variable (lnwg) is log hourly wages in current job. The main independent variable (hchild) is total number of children the respondent had at the time of the interview. Other variables in the model include whether or not married (mar) or divorced (div); educational attainment (eduatt); currently in school (cursc); several measures of part-time and full-time work experience (snrpt, snrft, exppt and expft); and breaks in

employment history (break). See Bollen and Brand for more details on the variables and sample selection.

The data set is strongly balanced, but several cases have missing data on one or more variables. Also, it is a fixed effects model but includes time invariant variables. Here we compare the results from `xtdpdml` and `xtabond`. First we give the code and then excerpts from the output

```
*** Section 4.2 -- Comparisons with A-B, real data
* Using fiml
use http://www3.nd.edu/~rwilliam/statafiles/bollenbrand, clear
set matsize 7500
set more off
xtabond lnwg hchild marr div eduatt cursc snrpt snrft exppt expft break black hisp
estimates store gmm
xtdpdml lnwg hchild marr div eduatt cursc snrpt snrft exppt expft break , ///
       constinv errorinv fiml tfix store(fiml) ///
       inv(black hisp) ti(Adapted from Bollen & Brand Social Forces 2010)
* Using listwise deletion
xtdpdml lnwg hchild marr div eduatt cursc snrpt snrft exppt expft break , ///
      constinv errorinv tfix store(listwise) ///
      inv(black hisp) ti(Adapted from Bollen & Brand Social Forces 2010)
esttab gmm fiml_h listwise_h, mtitles(gmm fiml listwise) scalar(N_g)
```

The output includes:

```
. xtabond lnwg hchild marr div eduatt cursc snrpt snrft exppt expft break black hisp
note: black dropped from div() because of collinearity
note: hisp dropped from div() because of collinearity

Arellano-Bond dynamic panel-data estimation     Number of obs     =      8,915
Group variable: id                              Number of groups  =      3,488
Time variable: year
                                                Obs per group:
                                                              min =          1
                                                              avg =   2.555906
                                                              max =          4

Number of instruments =     21                  Wald chi2(11)     =    3315.32
                                                Prob > chi2       =     0.0000
One-step results
------------------------------------------------------------------------------
        lnwg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        lnwg |
         L1. |  -.0072789   .0402023    -0.18   0.856     -.086074    .0715163
             |
      hchild |  -.0091342   .0090602    -1.01   0.313    -.0268919    .0086236
        marr |   .0468352   .0168387     2.78   0.005     .0138321    .0798384
         div |   .0747365   .0225606     3.31   0.001     .0305184    .1189545
      eduatt |   .0575892   .0102432     5.62   0.000     .0375128    .0776656
       cursc |   -.081103   .0153101    -5.30   0.000    -.1111103   -.0510956
       snrpt |   .0132922   .0054544     2.44   0.015     .0026018    .0239826
       snrft |   .0140817   .0027054     5.21   0.000     .0087792    .0193842
       exppt |    .056597   .0055597    10.18   0.000     .0457002    .0674937
       expft |   .0608082    .004636    13.12   0.000     .0517219    .0698946
       break |   .0200741   .0069791     2.88   0.004     .0063953    .0337528
       black |          0  (omitted)
        hisp |          0  (omitted)
       _cons |   .6280031   .1360759     4.62   0.000     .3612993     .894707
------------------------------------------------------------------------------
Instruments for differenced equation
        GMM-type: L(2/.).lnwg
        Standard: D.hchild D.marr D.div D.eduatt D.cursc D.snrpt D.snrft
                  D.exppt D.expft D.break
Instruments for level equation
        Standard: _cons
```

```
. xtdpdml lnwg hchild marr div eduatt cursc snrpt snrft exppt expft break , ///
>        constinv errorinv fiml tfix store(fiml) ///
>        inv(black hisp) ti(Adapted from Bollen & Brand Social Forces 2010)

Highlights: Adapted from Bollen & Brand Social Forces 2010
------------------------------------------------------------------------------
             |                 OIM
        lnwg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
lnwg         |
        lnwg |
         L1. |   .3378925   .0124879    27.06   0.000     .3134166    .3623684
             |
      hchild |  -.0209521   .0063606    -3.29   0.001    -.0334186   -.0084856
        marr |   .0359839    .012841     2.80   0.005     .0108161    .0611517
         div |   .0617287    .017081     3.61   0.000     .0282505     .095207
      eduatt |   .0583252   .0072068     8.09   0.000     .0442001    .0724503
       cursc |  -.1075845   .0132218    -8.14   0.000    -.1334988   -.0816701
       snrpt |   .0088462   .0043731     2.02   0.043     .0002751    .0174173
       snrft |   .0174143   .0021041     8.28   0.000     .0132904    .0215383
       exppt |   .0308717   .0037348     8.27   0.000     .0235517    .0381917
       expft |   .0307015   .0022474    13.66   0.000     .0262966    .0351064
       break |   .0370938   .0043345     8.56   0.000     .0285983    .0455893
       black |  -.0074612   .0103375    -0.72   0.470    -.0277223    .0127999
        hisp |   .0730661    .012675     5.76   0.000     .0482236    .0979086
------------------------------------------------------------------------------
# of units = 5285. # of periods = 6. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(218)  =     831.04, Prob > chi2 =  0.0000
IC Measures: BIC =  345115.17  AIC =  334921.02
Wald test of all coeff = 0: chi2(13) =    6701.90, Prob > chi2 =  0.0000
```

The results are strikingly different. Almost 21,000 records have data on at least one variable in the model, and all of these observations are used by `xtdpdml`(with the `fiml` option). However, only 8,915 records are used by `xtabond` because it deletes any record with missing data. Perhaps for this reason, `xtabond` produces a highly implausible estimate of almost zero effect of lagged wages on current wages and also says that the effect of the main independent variable, number of children, is statistically insignificant. In the `xtdpdml` results both effects are highly significant and the signs of the effects are in the expected direction. Many other variables have larger *z*-statistics in `xtdpdml` than they do in `xtabond`. `xtabond` cannot estimate effects for the time-invariant variables black and hisp, while `xtdpdml` shows that the effect of hisp is highly significant.

Even if we leave out the `fiml` option, thereby deleting all persons who have missing data at any time point, the results from `xtdpdml` seem somewhat more plausible. As we would expect, the smaller sample size causes effects to be less statistically significant. But, the effect of lagged wages continues to be positive and highly significant.

```
. * Using listwise deletion
. xtdpdml lnwg hchild marr div eduatt cursc snrpt snrft exppt expft break , ///
>       constinv errorinv tfix store(listwise) ///
>       inv(black hisp) ti(Adapted from Bollen & Brand Social Forces 2010)

Highlights: Adapted from Bollen & Brand Social Forces 2010
------------------------------------------------------------------------------
             |                 OIM
        lnwg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
lnwg         |
        lnwg |
         L1. |   .2776779   .0171863    16.16   0.000     .2439933    .3113624
             |
      hchild |  -.0144888   .0099824    -1.45   0.147     -.034054    .0050765
        marr |   .0590624   .0186809     3.16   0.002     .0224486    .0956763
         div |   .0577631   .0253645     2.28   0.023     .0080495    .1074767
      eduatt |   .0763836   .0108579     7.03   0.000     .0551024    .0976647
       cursc |  -.0923283   .0191093    -4.83   0.000    -.1297819   -.0548747
       snrpt |   .0150824   .0059555     2.53   0.011     .0034099    .0267549
       snrft |   .0101602   .0027619     3.68   0.000      .004747    .0155734
       exppt |     .04097   .0054855     7.47   0.000     .0302186    .0517215
       expft |   .0379746   .0030576    12.42   0.000     .0319817    .0439675
       break |   .0195759   .0075289     2.60   0.009     .0048195    .0343322
       black |  -.0299847    .017994    -1.67   0.096    -.0652523    .0052829
        hisp |   .0737694   .0220227     3.35   0.001     .0306056    .1169332
------------------------------------------------------------------------------
# of units = 1229. # of periods = 6. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(218)  =     612.60, Prob > chi2 =  0.0000
IC Measures: BIC =  113801.75  AIC =  105870.00
Wald test of all coeff = 0: chi2(13) =    3171.73, Prob > chi2 =  0.0000
```

In fairness, `xtdpdml` took far longer to run than did `xtabond`. Further, there will be other situations where the two methods will yield more similar results; and, when panels are far from being strongly balanced, `xtabond` may work much better (or `xtdpdml` may not work at all). But, at least in this particular case, where many cases have missing data and time-invariant variables are in the model, `xtdpdml` clearly seems to be the superior alternative.

As a sidelight, because we used the `store` option with `xtdpdml` and the `estimates store` command after `xtabond`, we can use a command like `esttab` (available from SSC) to present a table of the results. We should use the highlights version of the `xtdpdml` stored results, in this case fiml_h and listwise_h.

```
. esttab gmm fiml_h listwise_h, mtitles(gmm fiml listwise) scalar(N_g) z

------------------------------------------------------------
                      (1)             (2)             (3)
                      gmm            fiml        listwise
------------------------------------------------------------
main
L.lnwg           -0.00728          0.338***        0.278***
                  (-0.18)         (27.06)         (16.16)

hchild           -0.00913         -0.0210***      -0.0145
                  (-1.01)         (-3.29)         (-1.45)

marr              0.0468**         0.0360**        0.0591**
                  (2.78)           (2.80)          (3.16)

div               0.0747***        0.0617***       0.0578*
                  (3.31)           (3.61)          (2.28)

eduatt            0.0576***        0.0583***       0.0764***
                  (5.62)           (8.09)          (7.03)

cursc            -0.0811***       -0.108***       -0.0923***
                 (-5.30)          (-8.14)         (-4.83)

snrpt             0.0133*         0.00885*         0.0151*
                  (2.44)           (2.02)          (2.53)

snrft             0.0141***        0.0174***       0.0102***
                  (5.21)           (8.28)          (3.68)

exppt             0.0566***        0.0309***       0.0410***
                 (10.18)           (8.27)          (7.47)

expft             0.0608***        0.0307***       0.0380***
                 (13.12)          (13.66)         (12.42)

break             0.0201**         0.0371***       0.0196**
                  (2.88)           (8.56)          (2.60)

black                0           -0.00746         -0.0300
                    (.)           (-0.72)         (-1.67)

hisp                 0             0.0731***       0.0738***
                    (.)           (5.76)          (3.35)

_cons             0.628***
                  (4.62)
------------------------------------------------------------
N                  8915            5285            1229
N_g                3488            5285            1229
------------------------------------------------------------
z statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001
```

Of course, since we do not know the true population parameters, we also do not know for sure which approach actually produced better estimates. The next example therefore examines how well the Full Information Maximum Likelihood approach works when there are missing data.

4.3. Missing data. `xtabond` does listwise deletion by record (person-time) while `xtdpdml` does listwise deletion by panel (person), i.e., all the data for a person will be deleted if even a single variable at a single time has a missing value. Consequently, `xtdpdml` in its default mode can potentially delete much more data than `xtabond`. Full Information Maximum Likelihood (Arbuckle 1996) can often be very effective for dealing with data that are missing at random. It is generally much easier to use the `fiml` option than it is to use multiple imputation. In the example below we first present results using complete data. We then randomly set some values to missing, causing over half the cases to be lost because of listwise deletion of missing data. Even though data were randomly set to missing, the results are quite different from the original non-missing data. We then use `fiml`, producing results almost identical to those when there is no missing data. The `dec(3)` option limits the number of decimal places dislayed, making the output easier to read.

```
. * Results with no missing data -- provides a baseline for
. * assessing how well fiml works.
. use http://www3.nd.edu/~rwilliam/statafiles/wages, clear
. xtset id t
       panel variable:  id (strongly balanced)
        time variable:  t, 1 to 7
                delta:  1 unit

. xtdpdml wks L.lwage, inv(ed) pre(L.union) ti(Baseline with no missing data)
sto(nomissing) dec(3)

Highlights: Baseline with no missing data
------------------------------------------------------------------------------
             |                 OIM
         wks |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
wks          |
         wks |
          L1. |      0.187      0.020     9.27   0.000        0.148       0.227
             |
       lwage |
          L1. |      0.642      0.484     1.33   0.185       -0.307       1.591
             |
       union |
          L1. |     -1.191      0.517    -2.30   0.021       -2.204      -0.178
             |
          ed |     -0.112      0.056    -2.01   0.045       -0.222      -0.003
------------------------------------------------------------------------------
# of units = 595. # of periods = 7. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(71)  =     110.23, Prob > chi2 =  0.0020
IC Measures: BIC =   25470.43  AIC =   24772.64
Wald test of all coeff = 0: chi2(4) =      90.09, Prob > chi2 =  0.0000

. * Now we randomly create MD since there is none. But normally you
. * would not do this!
. replace union = . if _n/10 == int(_n/10)
(416 real changes made, 416 to missing)
```

```
. * fiml not used -- 60% of cases lost, estimates are quite a bit off.
. xtdpdml wks L.lwage, inv(ed) pre(L.union) ti(Baseline with missing data, no fiml)
sto(nofiml) dec(3)

Highlights: Baseline with missing data, no fiml
------------------------------------------------------------------------------
             |                 OIM
         wks |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
wks          |
         wks |
          L1. |      0.248      0.033     7.55   0.000        0.184       0.313
             |
       lwage |
          L1. |      1.613      0.869     1.86   0.063       -0.090       3.316
             |
       union |
          L1. |     -0.346      0.866    -0.40   0.689       -2.043       1.350
             |
           ed |     -0.208      0.087    -2.39   0.017       -0.379      -0.037
------------------------------------------------------------------------------
# of units = 238. # of periods = 7. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(71)  =     167.22, Prob > chi2 =  0.0000
IC Measures: BIC =   10558.77  AIC =   10006.68
Wald test of all coeff = 0: chi2(4) =      62.63, Prob > chi2 =  0.0000
```

Even though missing data were generated at random, there are clear differences both in the estimated coefficients and in the significance levels.

```
. * fiml used -- works extremely well, at least in this case
. xtdpdml wks L.lwage, inv(ed) pre(L.union) fiml ti (Baseline with missing data &
fiml) sto(fiml) dec(3)

Highlights: Baseline with missing data & fiml
------------------------------------------------------------------------------
             |                 OIM
         wks |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
wks          |
         wks |
          L1. |      0.187      0.020     9.26   0.000        0.148       0.227
             |
       lwage |
          L1. |      0.651      0.485     1.34   0.179       -0.299       1.602
             |
       union |
          L1. |     -1.181      0.557    -2.12   0.034       -2.273      -0.090
             |
           ed |     -0.112      0.057    -1.97   0.049       -0.224      -0.000
------------------------------------------------------------------------------
# of units = 595. # of periods = 7. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(71)  =     111.85, Prob > chi2 =  0.0014
IC Measures: BIC =   25680.69  AIC =   24982.91
Wald test of all coeff = 0: chi2(4) =      89.16, Prob > chi2 =  0.0000
```

With fiml used, both the coefficients and the standard errors are almost indistinguishable from the earlier results when no data were missing.

We can again use a command like `esttab` to present a convenient table of results. Note that `esttab` has an option for computing and including BIC and AIC statistics in its tables; but

because highlights files lack the information `esttab` needs to compute BIC and AIC correctly, we instead use the BIC and AIC scalars that are ereturned by `xtdpdml`.

```
. esttab nomissing_h nofiml_h fiml_h, mtitles(NoMissing NoFiml Fiml) scalar(chi2_ms
df_ms p_ms BIC AIC ) z
```

```
------------------------------------------------------------
                      (1)             (2)             (3)
                NoMissing          NoFiml            Fiml
------------------------------------------------------------
wks
L.wks              0.187***        0.248***        0.187***
                  (9.27)          (7.55)          (9.26)

L.lwage            0.642           1.613           0.651
                  (1.33)          (1.86)          (1.34)

L.union           -1.191*         -0.346          -1.181*
                 (-2.30)         (-0.40)         (-2.12)

ed                -0.112*         -0.208*         -0.112*
                 (-2.01)         (-2.39)         (-1.97)
------------------------------------------------------------
N                    595             238             595
chi2_ms            110.2           167.2           111.9
df_ms                 71              71              71
p_ms             0.00198        9.81e-10         0.00143
BIC              25470.4         10558.8         25680.7
AIC              24772.6         10006.7         24982.9
------------------------------------------------------------
z statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001
```

4.4. Panel Model with fixed effects; Goodness of Fit measures. In their 2010 Social Forces paper, Bollen and Brand present a series of Panel Models with Random and Fixed Effects. They used MPlus; but now, many, perhaps all, of their models can be easily replicated with `xtdpdml` (although hand tweaking of the code may be required in a few cases). In this relatively simple example there are no lagged independent variables. Here we present the fixed effects model 2 from their Table 3. We also include the `gof` option, which causes several goodness of fit measures to be included in the output.

```
. * Bollen & Brand Social Forces 2010 Fixed Effects Table 3 Model 2 p. 15
. use http://www3.nd.edu/~rwilliam/statafiles/bollenbrand, clear
(Bollen & Brand 2010 Social Forces V 89(1) NLSY 1983-1993 Odd years Long format)
```

```
. xtdpdml lnwg hchild marr div, ylag(0) fiml tfix errorinv gof sto(baseline)

Highlights: Dynamic Panel Data Model using ML for outcome variable lnwg
------------------------------------------------------------------------------
             |                 OIM
        lnwg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
lnwg         |
      hchild |  -.0704949   .0055935   -12.60   0.000     -.081458   -.0595319
        marr |   .0826099   .0104827     7.88   0.000     .0620642    .1031556
         div |   .0572981    .014612     3.92   0.000     .0286591    .0859372
------------------------------------------------------------------------------
# of units = 5231. # of periods = 6. First dependent variable is from period 1.
LR test of model vs. saturated: chi2(106)  =    1940.93, Prob > chi2 =  0.0000
IC Measures: BIC =   73683.21  AIC =   72252.61
Wald test of all coeff = 0: chi2(3) =     204.34, Prob > chi2 =  0.0000

------------------------------------------------------------------------------
Fit statistic        |      Value   Description
---------------------+--------------------------------------------------------
Likelihood ratio     |
        chi2_ms(106) |   1940.932   model vs. saturated
            p > chi2 |      0.000
        chi2_bs(123) |   8307.362   baseline vs. saturated
            p > chi2 |      0.000
---------------------+--------------------------------------------------------
Population error     |
               RMSEA |      0.058   Root mean squared error of approximation
 90% CI, lower bound |      0.055
         upper bound |      0.060
              pclose |      0.000   Probability RMSEA <= 0.05
---------------------+--------------------------------------------------------
Information criteria |
                 AIC |  72252.615   Akaike's information criterion
                 BIC |  73683.209   Bayesian information criterion
---------------------+--------------------------------------------------------
Baseline comparison  |
                 CFI |      0.776   Comparative fit index
                 TLI |      0.740   Tucker-Lewis index
---------------------+--------------------------------------------------------
Size of residuals    |
                  CD |      0.870   Coefficient of determination
------------------------------------------------------------------------------
Note: SRMR is not reported because of missing values.
```

See Acock (2013) for a discussion of goodness of fit measures in SEM. The "model vs. saturated" chi-square is a test of all 106 over-identifying restrictions implied by the model. Here the chi-square is almost 20 times its degrees of freedom, suggesting a poor fit to the data. However, as is well known, it is difficult to find any reasonably parsimonious model that will pass this test with a sample size of more than 5,000. Other GOF tests reported here are less sensitive to sample size. An RMSEA of less than .05 is considered to be a good fit, and we are almost there at .058. On the other hand, both the CLI and TLI are well below .90, the usual standard for a minimally acceptable model.

If we were unhappy with the model fit we could consider relaxing some of the constraints of the model, e.g. we could let the effects of some variables differ across time by using the `xfree` or `yfree` options. Modification indices (obtained with the `sem` postestimation command `estat mindices`) could provide additional guidance on how to modify the model. Because there are

so many equality constraints imposed by the model, the `estat scoretests` command may be especially useful, because it displays score tests (Lagrangian multiplier tests) for each of the user-specified linear constraints imposed on the model when it was fit. In this case,

```
. estat scoretests

Score tests for linear constraints

  ( 1)  [lnwg1]hchild1 - [lnwg6]hchild6 = 0
  ( 2)  [lnwg1]marr1 - [lnwg6]marr6 = 0
  ( 3)  [lnwg1]div1 - [lnwg6]div6 = 0
  ( 4)  [lnwg1]Alpha = 1
  ( 5)  [lnwg2]hchild2 - [lnwg6]hchild6 = 0
  ( 6)  [lnwg2]marr2 - [lnwg6]marr6 = 0
  ( 7)  [lnwg2]div2 - [lnwg6]div6 = 0
  ( 8)  [lnwg2]Alpha = 1
  (12)  [lnwg3]Alpha = 1
  (13)  [lnwg4]hchild4 - [lnwg6]hchild6 = 0
  (15)  [lnwg4]div4 - [lnwg6]div6 = 0
  (16)  [lnwg4]Alpha = 1
  (17)  [lnwg5]hchild5 - [lnwg6]hchild6 = 0
  (20)  [lnwg5]Alpha = 1
  (21)  [lnwg6]Alpha = 1
  (22)  [var(e.lnwg1)]_cons - [var(e.lnwg6)]_cons = 0
  (23)  [var(e.lnwg2)]_cons - [var(e.lnwg6)]_cons = 0
  (24)  [var(e.lnwg3)]_cons - [var(e.lnwg6)]_cons = 0
--------------------------------------
             |      chi2     df  P>chi2
-------------+------------------------
        ( 1) |    54.652      1    0.00
        ( 2) |    17.970      1    0.00
        ( 3) |     4.194      1    0.04
        ( 4) |   543.286      1    0.00
        ( 5) |    15.011      1    0.00
        ( 6) |     5.726      1    0.02
        ( 7) |     8.885      1    0.00
        ( 8) |    91.101      1    0.00
        (12) |     5.594      1    0.02
        (13) |     5.866      1    0.02
        (15) |     4.213      1    0.04
        (16) |    98.223      1    0.00
        (17) |     4.062      1    0.04
        (20) |   100.611      1    0.00
        (21) |   134.406      1    0.00
        (22) |    12.887      1    0.00
        (23) |    20.007      1    0.00
        (24) |    20.581      1    0.00
--------------------------------------
```

The results strongly suggest that the Alpha coefficients (i.e. the coefficients for the unmeasured fixed effects; see tests 4, 8, 12, 16, 20 and 21) are not the same across time. The constraint that the error variances are the same across time (which was imposed by the `errorinv` option) also seems dubious (see tests 22, 23, and 24). We therefore relax those constraints by dropping the `errorinv` option and adding `alphafree`, resulting in

```
. xtdpdml lnwg hchild marr div, ylag(0) fiml tfix alphafree gof sto(modified)
. lrtest baseline_f modified_f, stats

[Some output deleted]

------------------------------------------------------------------------------
Fit statistic        |      Value   Description
---------------------+--------------------------------------------------------
Likelihood ratio     |
         chi2_ms(96) |    789.252   model vs. saturated
            p > chi2 |      0.000
        chi2_bs(123) |   8307.362   baseline vs. saturated
            p > chi2 |      0.000
---------------------+--------------------------------------------------------
Population error     |
               RMSEA |      0.037   Root mean squared error of approximation
 90% CI, lower bound |      0.035
         upper bound |      0.040
              pclose |      1.000   Probability RMSEA <= 0.05
---------------------+--------------------------------------------------------
Information criteria |
                 AIC |  71120.934   Akaike's information criterion
                 BIC |  72617.152   Bayesian information criterion
---------------------+--------------------------------------------------------
Baseline comparison  |
                 CFI |      0.915   Comparative fit index
                 TLI |      0.891   Tucker-Lewis index
---------------------+--------------------------------------------------------
Size of residuals    |
                  CD |      0.899   Coefficient of determination
------------------------------------------------------------------------------
Note: SRMR is not reported because of missing values.

Likelihood-ratio test                                 LR chi2(10) =   1151.68
(Assumption: baseline_f nested in modified_f)         Prob > chi2 =    0.0000

Akaike's information criterion and Bayesian information criterion

------------------------------------------------------------------------------
       Model |        Obs  ll(null)  ll(model)      df         AIC        BIC
-------------+----------------------------------------------------------------
  baseline_f |      5,231         .  -35908.31     218    72252.61   73683.21
  modified_f |      5,231         .  -35332.47     228    71120.93   72617.15
------------------------------------------------------------------------------
               Note: N=Obs used in calculating BIC; see [R] BIC note.
```

Relaxing the constraints on Alpha and the error variances requires only 10 degrees of freedom and produces a dramatic improvement in model fit. The BIC and AIC statistics also strongly favor the less constrained model. There are also clear improvements in other goodness of fit measures. RMSEA = .037, much better than the .05 value that is considered a good fit. Similarly CLI = .891 and TLI = .915, close to or better than the usual standard of .90 for a minimally acceptable model.

Of course the researcher may also want to reconsider whether other model assumptions (such as not including any lagged independent variables, especially lagged *y*) are justified.

4.5 Fixed Effects Versus Random Effects Models; Plus, An Alternative to the Hausman Test. Allison (2009) notes that a Hausman test is often used to contrast fixed effects and random

effects models. He notes, however, that the Hausman test can sometimes be problematic, e.g. it can produce negative values for some data configurations. He argues that a likelihood ratio test can have superior statistical properties. He originally estimated his models using MPlus (Stata `sem` did not exist when he wrote his book) but the results presented in his Table 6.1 can now be easily reproduced using `xtdpdml`. He analyzes data from the National Longitudinal Survey of Youth for 581 children who were interviewed in 1990, 1992, and 1994. The dependent variable (anti) is a measure of antisocial behavior (higher scores indicate higher levels of antisocial behavior), while time-varying measures of poverty (pov) and self-esteem (self) are independent variables. There are also several time-invariant independent variables, including measures for race and ethnicity (black and hispanic), respondent's age at time of first interview (childage), whether or not the respondent's mother was married in 1990 (married), the respondent's gender (gender; 1= female, 0 = male), the respondent's mother's age when the respondent was born (momage), and whether or not the respondent's mother was employed in 1990. See Allison (2009) for more information on the data and sample selection.

```
* Replicate Allisons Table 6.1 using xtdpdml
use http://www3.nd.edu/~rwilliam/statafiles/nlsy, clear
gen id = _n
reshape long anti pov self, i(id) j(year)
xtset id year, delta(2)

* Table 6.1 - Random effects
xtdpdml anti self pov, inv(black hispanic childage married gender momage momwork) ///
       ylag(0) tfix errorinv re store(re)


* Table 6.1 - Fixed effects
xtdpdml anti self pov, inv(black hispanic childage married gender momage momwork) ///
       ylag(0) tfix errorinv store(fe)

esttab re_h fe_h , mtitles(Random Fixed) scalar(chi2_ms df_ms p_ms BIC AIC ) z
lrtest re_f fe_f, stats
```

Showing just the last part of the output,

```
. esttab re_h fe_h , mtitles(Random Fixed) scalar(chi2_ms df_ms p_ms BIC AIC ) z

--------------------------------------------
                      (1)             (2)
                   Random           Fixed
--------------------------------------------
anti
self              -0.0621***      -0.0552***
                  (-6.54)         (-5.25)

pov                 0.247**         0.112
                   (3.07)          (1.21)

black               0.227           0.269*
                   (1.81)          (2.13)

hispanic           -0.218          -0.198
                  (-1.59)         (-1.43)

childage           0.0885          0.0895
                   (0.98)          (0.98)

married           -0.0496         -0.0221
                  (-0.39)         (-0.17)

gender             -0.483***       -0.476***
                  (-4.56)         (-4.47)

momage            -0.0219         -0.0255
                  (-0.87)         (-1.01)

momwork             0.261*          0.296*
                   (2.29)          (2.57)
--------------------------------------------
N                     581             581
chi2_ms             84.56           66.57
df_ms                 34              28
p_ms           0.00000341      0.0000554
BIC               24248.7         24268.9
AIC               23733.6         23727.6
--------------------------------------------
z statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001

. lrtest re_f fe_f, stats

Likelihood-ratio test                                 LR chi2(6)  =     18.00
(Assumption: re_f nested in fe_f)                     Prob > chi2 =    0.0062

Akaike's information criterion and Bayesian information criterion

-----------------------------------------------------------------------------
       Model |        Obs  ll(null)  ll(model)      df         AIC        BIC
-------------+---------------------------------------------------------------
        re_f |        581         .  -11748.82     118    23733.63   24248.67
        fe_f |        581         .  -11739.82     124    23727.63   24268.86
-----------------------------------------------------------------------------
               Note: N=Obs used in calculating BIC; see [R] BIC note.
```

Substantively, the most striking differences between the two models are that, in the fixed effects model, the effects of self-esteem and poverty are smaller than they are in the random effects model; indeed in the fixed effects model the effect of poverty is not statistically significant. This

presumably reflects the fact that the fixed effects model better controls for unobservable variables not in the model than does the random effects model.

Based on the chi-square test and the AIC statistics, we favor the fixed effects model over the random effects model. (However if the BIC statistics are used instead, the more parsimonious random effects model would be favored.) Note that, with more conventional fixed effects methods, the effects of time-invariant variables (of which there are several in this case) cannot be estimated. Luckily, this is not the case with the `xtdpdml` approach.

4.6 Reciprocal Causation. Allison (2009) notes that we are often interested in knowing whether *x* causes *y*, *y* causes *x*, or both. That is, we want to know whether reciprocal causation is present. He therefore presents models with reciprocal effects and lagged predictors. He further notes that simulation studies (Allison 2000) have shown that SEM approaches do an excellent job of recovering the parameters for such models. While there are different SEM approaches for estimating such models, he argues that estimating each equation separately allows for considerably more flexibility in specifying the model. He analyzes data for 178 occupations from the March Current Population Surveys for 1983, 1989, 1995, and 2001. For each occupation he computed the proportion female (pf) and the median wage for females (mdwgf). He only analyzed occupations which had at least 50 sample members in each year. Allison originally estimated his models using MPlus, but the results presented in his table 6.3 can now be easily reproduced using `xtdpdml`.

```
* Replicate Allison's Table 6.3 using xtdpdml
use http://www3.nd.edu/~rwilliam/statafiles/occ, clear
gen id = _n
reshape long pf mdwgf, i(id) j(j)
xtset id j
* Median Wage
xtdpdml mdwgf, pre(L.pf) store(mdwgf)
* Proportion female
xtdpdml pf, pre(L.mdwgf) store(pf)
```

Showing the results from the `xtdpdml` commands,

```
. * Median Wage
. xtdpdml mdwgf, pre(L.pf) store(mdwgf)

Highlights: Dynamic Panel Data Model using ML for outcome variable mdwgf
------------------------------------------------------------------------------
             |                 OIM
       mdwgf |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
mdwgf        |
       mdwgf |
         L1. |   .3439355   .0637916     5.39   0.000     .2189064    .4689647
             |
          pf |
         L1. |  -.1587604   2.447221    -0.06   0.948    -4.955226    4.637705
------------------------------------------------------------------------------
# of units = 178. # of periods = 4. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(7)  =      13.82, Prob > chi2 =  0.0546
IC Measures: BIC =    2743.07  AIC =    2653.98
Wald test of all coeff = 0: chi2(2) =      29.48, Prob > chi2 =  0.0000
```

```
. * Proportion female
. xtdpdml pf, pre(L.mdwgf) store(pf)

Highlights: Dynamic Panel Data Model using ML for outcome variable pf
------------------------------------------------------------------------------
             |                 OIM
          pf |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
pf           |
          pf |
         L1. |    .299357   .0793574     3.77   0.000     .1438194    .4548946
             |
       mdwgf |
         L1. |  -.0005521    .001515    -0.36   0.716    -.0035214    .0024172
------------------------------------------------------------------------------
# of units = 178. # of periods = 4. First dependent variable is from period 2.
LR test of model vs. saturated: chi2(7)  =      20.40, Prob > chi2 =  0.0048
IC Measures: BIC =    1492.45  AIC =    1403.36
Wald test of all coeff = 0: chi2(2) =      14.25, Prob > chi2 =  0.0008
```

Allison (2009, p. 96) notes that

> Not surprisingly, each variable has a positive, statistically significant effect on itself six years later. With respect to the "cross lagged" coefficients, however, there is no evidence for an effect in either direction.

Based on this and other analyses he conducts, Allison rejects the idea that reciprocal causation is present.

5. Special Topics

The `xtdpdml` approach requires that some problems be approached differently than they are with other models. In addition, for some applications, `xtdpdml` models can be harder to estimate and may consume far more computing time than other approaches. In this section we discuss various ways to deal with these issues.

5.1 Interactions with Time

Researchers sometimes want constants and variable effects to differ across time. `xtdpdml` can do this but, because data are reshaped wide, the procedure is different than it is with other programs.

By default, `xtdpdml` lets the constants differ across time periods. In other programs this would be like including i.time in the model. The `constinv` or `nocsd` options can be specified if the user wants the constants to be invariant across time. Note that using these options will sometimes cause convergence problems.

In other situations the user might want interactions with time where the effect of a variable is free to differ across time periods. In other programs this might be accomplished by specifying something like i.time#c.ses. With `xtdpml` you use the free options instead, e.g. `xfree(ses)`

will allow the effect of ses to differ at each time period. Similarly, `alphafree` might be used if it was thought that the fixed effects differed across time.

5.2 Speed, Convergence, and Missing Data Problems

`xtdpdml` sometimes has trouble converging to a solution or else is extremely slow in doing so. This might occur, for example, when time-varying variables do not vary that much across time, creating problems of collinearity in estimation. Here are some things you can try when that happens.

Stata 14.2 introduced major enhancements to the `sem` command that dramatically helped with both convergence and speed, especially when full information maximum likelihood is used. Try to use 14.2 or later when using `xtdpdml`.

By default, `sem` deletes cases on a listwise basis. Because data are converted to wide format, a missing wave or even missing data on a single variable at a single time can cause all the data for a case to be lost. In addition `xtdpdml` models are computationally intensive. `xtdpdml` therefore works best when panels are strongly balanced, *T* is small (e.g. less than 10), and there are no missing data. If these conditions do not apply to your data, consider doing the following.

- The `fiml` option will often help when some data are missing or when entire waves are missing for some individuals. Nonetheless, while fiml worked very well in the examples presented here, we have found it can have problems with extremely unbalanced panels, especially when some time periods have only a few cases.

- Consider restricting your data to a smaller range of time periods where most or all cases have complete data. Or, you might consider using only every *k*th year, e.g. 1980, 1985, 1990, ..., 2015. Using fewer variables in the model may also help.

- Consider rescaling variables, e.g. measure income in thousands of dollars rather than in dollars. This can help with numerical precision problems. The `std` option makes rescaling and standardizing variables easy, although it may make coefficients a little harder to interpret. If `std` solves a convergence problem then you may want to rescale the variables yourself in a more interpretable way, e.g. if income is measured in dollars then compute income/1000 to measure income in thousands of dollars.

- Stata 14.2 changed the way start values are computed. Our experience is that models using fiml tend to run far more quickly in 14.2 compared with earlier versions. However, sometimes the new start values actually make the models run more slowly or cause convergence problems. If you are running Stata 14.2 or later, you can add the options `skipcfatransform` and/or `skipconditional` to make Stata use the old starting values method. `altstart` is an easy way to specify both options.

- Mplus sometimes, but albeit not always, succeeds when Stata has problems and is often much faster. Try the `mplus` option if you have access to the program.

There are several other options you can try if you are having problems achieving convergence. Much of this advice applies to many programs, not just `xtdpdml`.

- The `difficult` option will sometimes work miracles. There is no guarantee it will work (sometimes it makes things worse) but it is very easy to try.

- The `technique` option can be specified to use different maximization techniques. See the help for maximize.

- `evars` sometimes helps with convergence when there are no predetermined variables in the model. It is an alternative and usually less efficient way of specifying the error terms. But sometimes it helps and may be necessary for replicating results from earlier versions of `xtdpdml`.

- The `iterate` option can be used to increase or decrease the number of iterations `xtdpdml` tries before giving up. The `details` option will show the iteration log. You can increase or decrease the number of iterations depending on whether it appears the program is converging to a solution.

Finally, remember that problems with regressing *y* on lagged *y* are not that severe when *N* is large and/or *T* is large and/or the autoregressive coefficient is small. Commands like `xtreg` or `xtabond` may meet your needs in such situations. But even then, as our examples showed, features like fiml and time-invariant independent variables may make it worth your while to pare your dataset down so you can do at least some analyses with `xtdpdml`.

6. Other alternatives to xtdpdml

The user-written routines `xtmoralb` (Moral-Benito 2013) and `xtdpdqml` (Kripfganz, 2015; available from SSC) can do some of the same things as `xtdpdml`, and may be useful in many situations. However, they also have some important limitations. `xtmoralb` works extremely well with predetermined variables (indeed we used it to refine `xtdpdml`). However, it cannot handle time-invariant variables, lagged exogenous variables, and `is` not fully efficient with strictly exogenous variables.

`xtdpdqml` works with strictly exogenous variables and can also sometimes produce results very similar to `xtdpdml`. However, it cannot handle time-invariant variables (in a fixed effects model) and (according to the author) is inappropriate for predetermined variables. Also, `xtdpdqml` implements the ml method of Hsiao et al (2002) which makes strong and questionable assumptions about initial conditions

7. Support

Additional information on the `xtdpdml` command, as well as suggestions for dealing with possible problems, can be found on its support page at

http://www3.nd.edu/~rwilliam/dynamic/index.html

8. Acknowledgments

Ken Bollen and Jennie Brand graciously provided us with the data from their 2010 Social Forces paper to use in our examples. UCLA and Michael Mitchell kindly allowed us to take their `stata2mplus` program and adapt it for our purposes. Code from Mead Over's `linewrap` program was modified for use with the `semfile` option. William Lisowski and Clyde Schechter provided comments that improved program coding. Paul von Hippel offered helpful comments on the program's documentation. Kristin MacDonald and other Stata Corp staff were very helpful in modifying Stata so that `sem` and `xtdpdml` would execute much more quickly.

9. References

Acock, A. C. 2013. *Discovering Structural Equation Modeling Using Stata, Revised ed.* College Station, TX: Stata Press.

Ahn, S. C. and Peter Schmidt (1995) "Efficient Estimation of Models for Dynamic Panel Data." *Journal of Econometrics* 68: 5-27.

Allison, Paul D. (2000) "Inferring Causal Order from Panel Data." Paper presented at the Ninth International Conference on Panel Data, June 22, Geneva, Switzerland.

Allison, Paul D. (2009) *Fixed Effects Regression Models*. Thousand Oaks, CA: Sage Publications.

Allison, Paul D. 2015. "Don't Put Lagged Dependent Variables in Mixed Models." http://statisticalhorizons.com/lagged-dependent-variables. Last accessed October 18, 2016.

Allison, Paul D. In Progress. "Maximum Likelihood for Dynamic Panel Models with Cross-Lagged Effects".

Anderson, T., Kunitomo, N. and Sawa, T. 1982. "Evaluation of the Distribution Function of the Limited Information Maximum Likelihood Estimator." *Econometrica*, 50: 1009–1027.

Arbuckle, James L. "Full information estimation in the presence of incomplete data". *Advanced structural equation modeling: Issues and techniques* 243 (1996): 277.

Arellano, M. and S. Bond (1991) "Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations." *The Review of Economic Studies* 58: 277-297.

Bai, Jushan. 2013. “Fixed-effects dynamic panel models, a factor analytical method.” *Econometrica*, Vol. 81, No. 1, 285–314.

Baltagi, Badi H. (2013), *Econometric Analysis of Panel Data.* Fifth Edition. New York: John Wiley & Sons.

Bollen, Kenneth, and Jennie Brand. 2010. “A General Panel Model with Random and Fixed Effects: A Structural Equations Approach.” *Social Forces* 89:1, 1-34.

Bun, M. and J. Kiviet. 2006. “The effects of dynamic feedbacks on LS and MM estimator accuracy in panel data models.” *Journal of Econometrics* 132: 409-444.

Cornwell, Christopher and Peter Rupert (1988) “Efficient Estimation with Panel Data: An Empirical Comparison of Instrumental Variables Estimators.” *Journal of Applied Econometrics* 3: 149-155.

Hsiao, Cheng (2014) *Analysis of Panel Data*. Third Edition. London: Cambridge University Press.

Moral-Benito, Enrique. 2013. “Likelihood-based Estimation of Dynamic Panels with Predetermined Regressors.” *Journal of Business and Economic Statistics* 31:4, 451-472.

Moral-Benito, Enrique, Paul D. Allison and Richard Williams. In progress. “Dynamic Panel Data Modeling using Maximum Likelihood: An Alternative to Arellano-Bond.” http://www3.nd.edu/~rwilliam/dynamic/Benito_Allison_Williams.pdf. Last accessed October 18, 2016.

Wooldridge, Jeffrey M. (2010) *Econometric Analysis of Cross Section and Panel Data*. Cambridge, MA: MIT Press.

Appendix: The Simulation Program, xtdptml_simul

```
* Code from Williams, Allison, Moral-Benito paper
* Simulations Program
* Based on design 5 in table 1 of moral-benito (2013)
program xtdpdml_simul
	*** Section 4.1, Comparisons with A-B, Simulated data
	set more off
	clear all
	version 13.1, user

	* Default values: seed = 3, bylab = .75, bx = .25, t = 4, nreps = 10, N = 100
	syntax, [SEEDnum(integer 3) bylag(real .75) bx(real .25) t(integer 4) ///
		ONLYConverged nreps(integer 10) N(integer 100)]

	quietly {
		local N `n'
		* The following are currently fixed but could be changed.
		local b3 .5
		local b4 -0.17
		local g 1
		local h 1
		local f .67

		set matsize 10000
		* reps = the number of simulations
		local reps `nreps'
		set seed `seednum'

		mat rxtdpdml = J(`reps',2,.)
		mat rxtdpd = J(`reps',2,.)
		local it = 1
		while `it' <= `reps'{

			clear
			set obs `N' /* this is N (units in the panel */
			gen id=_n
			gen c=rnormal()*1.72

			* initial observations under mean stationarity: eqs (29) and (30)
page 457 Moral-Benito (2013)
			gen y1= ( (`bx'*`f'+(1-`b3')) / ((1-`b3')*(1-`bylag')-`bx'*`b4') )
* c + `g'*rnormal()
			gen x1= ( (`b4'+`f'*(1-`bylag')) / ((1-`b3')*(1-`bylag')-
`bx'*`b4') ) * c + `g'*rnormal()*2.56

			forval tnum = 2/`t' {
				local tlag = `tnum' - 1
				gen x`tnum'= `b3'*x`tlag' + `b4'*y`tlag' +`f'*c +
`g'*rnormal()*2.56
				gen y`tnum'= `bx'*x`tnum' + `bylag'*y`tlag' +`h'*c +
`g'*rnormal()
			}

			reshape long y x, i(id) j(t)
			xtset id t

			di "`it'"

			/* xtdpdml */
			capture xtdpdml y, pre(x) skipcfatransform skipconditional
constinv
```

```
                        local mlconverged = e(converged)
                        if e(converged)== 1{
                              mat bylag = _b[y2:y1]
                              mat bx = _b[y2:x2]
                              mat rxtdpdml[`it',1] = bylag
                              mat rxtdpdml[`it',2] = bx
                        }

                        /* xtdpd */
                        *Could also use xtdpd y L.y x, dgmmiv(x L.y , lag(1 ))
                        xtabond y, pre(x)
                        if "`onlyconverged'" == "" | `mlconverged' == 1 {
                              mat bylag = _b[L.y]
                              mat bx = _b[x]
                              mat rxtdpd[`it',1] = bylag
                              mat rxtdpd[`it',2] = bx
                        }

                        local it = `it' + 1
                  }

                  svmat rxtdpdml
                  svmat rxtdpd

                  gen biasyml = rxtdpdml1 - `bylag'
                  gen biasxml = rxtdpdml2 - `bx'

                  gen biasygmm = rxtdpd1 - `bylag'
                  gen biasxgmm = rxtdpd2 - `bx'

            }

            display
            display as result "N = `N', T = `t', bylag = `bylag', bx = `bx', `reps'
simulations
            sum biasyml biasygmm biasxml biasxgmm
            display
            quietly sum biasyml
            display "Root Mean Squared Error of lagged y ml  = " sqrt(r(mean)^2 + r(Var))
            quietly sum biasygmm
            display "Root Mean Squared Error of lagged y gmm = " sqrt(r(mean)^2 + r(Var))
            quietly sum biasxml
            display "Root Mean Squared Error of x ml         = " sqrt(r(mean)^2 + r(Var))
            quietly sum biasxgmm
            display "Root Mean Squared Error of x gmm        = " sqrt(r(mean)^2 + r(Var))
            display

end
```

The commands used to execute the simulations were

```
* Section 4.1 -- simulations comparing GMM with ML
* xtdpdml_simul needs to be in the Stata search path
* WARNING: These can take several hours to run!!!
* You can get similar results more quickly by changing
* nreps to 500.

* Vary sample size
xtdpdml_simul, nreps(5000) t(4) bylag(.75) bx(.25) n(100) seed(3)
xtdpdml_simul, nreps(5000) t(4) bylag(.75) bx(.25) n(500) seed(3)
xtdpdml_simul, nreps(5000) t(4) bylag(.75) bx(.25) n(1000) seed(3)
```

```
xtdpdml_simul, nreps(5000) t(4) bylag(.75) bx(.25) n(5000) seed(3)

* Let T = 10
xtdpdml_simul, nreps(5000) t(10) bylag(.75) bx(.25) n(100) seed(3)

* Smaller autoregressive effect
xtdpdml_simul, nreps(5000) t(4) bylag(.50) bx(.25) n(100) seed(3)

* Not in paper -- drop nonconverging simulations from xtabond
xtdpdml_simul, nreps(5000) t(4) bylag(.75) bx(.25) n(100) seed(3) onlyconverged
```

We ran 5,000 simulations each time, which produces more precise estimates but is extremely time consuming. Similar results can be achieved by only running 500 or 1,000 simulations.