The Stata Journal (yyyy) **vv**, Number ii, pp. 1–16

Simplifying the Estimation of Correlated Random Effects Models

Fernando Rios-Avila
Levy Economics Institute of Bard College
Annandale-on-Hudson, NY 12504
friosavi@levy.org

Abstract.

This paper introduces `cre`, a Stata prefix command designed to simplify the implementation of Correlated Random Effects (CRE) models, following the Mundlak (1978) approach, for a wide range of linear and nonlinear estimation commands. Standard Fixed Effects (FE) estimators, while consistent under unobserved heterogeneity, cannot identify coefficients for time-invariant variables. Standard Random Effects (RE) estimators can identify such coefficients but rely on the strong, often violated, assumption that individual effects are uncorrelated with regressors. CRE models offer a pragmatic middle ground, providing FE-equivalent estimates for time-varying coefficients in linear models while still allowing identification of time-invariant effects. Crucially, the CRE approach extends readily to many nonlinear models where FE estimators are complex, inconsistent (due to the incidental parameter problem), or unavailable. The cre command facilitates this by automatically generating the required group means of time-varying regressors and adding them to the specified model, handling both balanced and unbalanced panels correctly based on established methods (Wooldridge 2019). It integrates seamlessly with Stata's factor variables and post-estimation commands like margins, enhancing its utility for applied researchers. This paper details the theoretical underpinnings, contrasts the approach with alternatives, explains the command's syntax and features, provides empirical examples, and presents simulation results demonstrating its performance for nonlinear models.

Keywords: st0001, Mundlak approach, correlated random effects, panel data, nonlinear models, Stata, prefix command

1 Introduction

Panel data offers significant advantages for empirical research by allowing researchers to control for unobserved individual heterogeneity that remains constant over time. The two dominant approaches for analyzing such data are fixed effects (FE) and random effects (RE) models. However, both have limitations. FE models provide consistent estimates by eliminating time-invariant unobserved factors, but consequently, cannot estimate the effects of observed time-invariant variables (e.g., gender, race, baseline characteristics), which are often of substantive interest (Wooldridge 2010). RE models estimate effects for both time-varying and time-invariant variables but require the strong assumption that the unobserved individual-specific effects are uncorrelated with the explanatory variables—an assumption frequently questioned in practice (Wooldridge 2019). Violating this assumption leads to inconsistent RE estimates.

A third, less commonly implemented approach, offers a compelling alternative: Correlated Random Effects (CRE) models. Originating with Mundlak (1978) and further developed by Chamberlain (1982), the CRE framework explicitly models the correlation between the unobserved individual effects and the explanatory variables. Specifically, the Mundlak (1978) approach, which is the focus here, assumes the individual effect is a linear projection of the *individual means* of the time-varying covariates, plus a random component uncorrelated with the covariates. This specification achieves two key goals: (1) In linear models, it yields estimates for time-varying coefficients that are identical to the FE estimator, thus providing consistency even when the RE assumption fails. (2) It allows for the estimation of coefficients on time-invariant variables.

Perhaps the most significant advantage of the Mundlak CRE approach lies in its applicability to **nonlinear models** (e.g., probit, logit, tobit, Poisson). For many such models, FE estimation is either computationally intensive, suffers from the incidental parameter problem (leading to inconsistency as T remains small), or simply does not exist (Wooldridge 2019). The CRE approach provides a practical and consistent estimation strategy by augmenting the model specification with the means of time-varying covariates before applying the standard pooled (cross-sectional) estimator, typically with cluster-robust standard errors (Wooldridge 2010, 2019).

Despite these benefits, CRE models are not as widely used as FE or RE, partly due to perceived implementation hurdles and lack of readily available, flexible software tools. While Stata recently introduced a `cre` option for `xtreg` (as of StataNow, June 25, 2024), its use is restricted to linear models estimated by `xtreg`. Community-contributed commands like `mundlak` (Perales 2013) and `xthybrid` (Schunck and Perales 2017) exist; however, `mundlak` focuses on linear models, and `xthybrid` implements a related but distinct "hybrid model" within a generalized mixed-effects framework, which may add complexity compared to the direct Mundlak specification.

This paper introduces the `cre` command, a **prefix command** in Stata, designed to bridge this gap. Its primary contribution is to provide a simple, flexible, and unified framework for estimating Mundlak-style CRE models across a wide range of Stata estimation commands, both linear and nonlinear. Key features include:

- **Prefix Functionality:** `cre` works seamlessly with most Stata estimation commands (official and community-contributed).
- **Automatic Mean Generation:** It automatically identifies time-varying regressors, computes their individual-specific means, and adds them to the model specification.
- **Unbalanced Panel Support:** It correctly handles unbalanced panels by calculating means using only the available observations for each individual within the estimation sample, following the approach validated by Wooldridge (2019, see specific discussion in Section 2.1).
- **Integration:** It fully supports Stata's factor variables and integrates smoothly

with post-estimation commands `margins` for computing average partial effects (APEs).

This paper proceeds as follows: Section 2 reviews the theoretical framework for CRE models in linear and nonlinear settings, discussing unbalanced panels, standard errors, APE calculation, and interaction terms. Section 3 describes the syntax and usage of the `cre` command. Section 4 presents empirical examples comparing `cre` with alternative estimators for both linear and nonlinear models using a standard dataset. Section 5 provides Monte Carlo simulation evidence on the performance of the CRE approach for nonlinear models. Section 6 concludes. This paper focuses exclusively on static models; **dynamic panel models** (incorporating lagged dependent variables) present additional complexities (e.g., initial conditions problem (Wooldridge 2010)) and are outside the scope of the current `cre` command and this discussion.

2 Theoretical Framework

2.1 Correlated Random Effects Models - Linear Case

Consider a standard linear panel data model:

$$y_{i,t} = \beta_0 + x_{i,t}\beta_x + z_i\beta_z + \alpha_i + u_{i,t} \tag{1}$$

where $y_{i,t}$ is the outcome for individual i at time t, $x_{i,t}$ is a $1 \times K$ vector of time-varying explanatory variables, z_i is a $1 \times G$ vector of time-invariant variables, α_i is the unobserved time-invariant individual-specific effect, and $u_{i,t}$ is the idiosyncratic error term, assumed uncorrelated with $x_{i,t}$, z_i, and α_i for all t.

The standard RE estimator assumes $Cov(x_{i,t}, \alpha_i) = 0$ and $Cov(z_i, \alpha_i) = 0$. If this holds, RE is consistent and efficient. However, if $Cov(x_{i,t}, \alpha_i) \neq 0$, the RE estimator is inconsistent. The FE estimator addresses this by transforming the data (e.g., demeaning) to eliminate α_i, yielding consistent estimates for β_x. However, this transformation also eliminates z_i, making β_z unidentified.

The Mundlak (1978) CRE approach offers a solution by explicitly modeling the correlation between α_i and the time-varying covariates $x_{i,t}$. It assumes that the expectation of α_i conditional on the individual means of $x_{i,t}$ is linear:[1]

$$E[\alpha_i|\bar{x}_i] = \gamma_0 + \bar{x}_i\gamma$$

where $\bar{x}_i = T_i^{-1}\sum_{t=1}^{T_i} x_{i,t}$ is the $1 \times K$ vector of individual-specific means of the time-varying variables for individual i over the T_i periods they are observed. This allows us

[1] An alternative approach to Mundlak's is the one proposed by Chamberlain (1982), which allows for a more flexible specification of the individual effects, where all realizations of the time-varying covariates are included in the model. This approach is more complex and computationally intensive, especially for unbalanced panels, and is not implemented in the `cre` command.

to write α_i as:

$$\alpha_i = \gamma_0 + \bar{x}_i\gamma + v_i \tag{2}$$

where v_i is a random component defined such that $E[v_i|\bar{x}_i] = 0$. More explicitly, the Mundlak approach assumes that v_i is uncorrelated with the *full history* of covariates $x_i = (x_{i,1}, ..., x_{i,T_i})$, not just the mean $\bar{x}_i$. Substituting Equation 2 into Equation 1 yields the CRE model specification:

$$y_{i,t} = (\beta_0 + \gamma_0) + x_{i,t}\beta_x + z_i\beta_z + \bar{x}_i\gamma + v_i + u_{i,t} \tag{3}$$

This augmented model can be estimated using pooled OLS or, more appropriately, using a RE estimator on the augmented specification. The composite error term $\mu_{i,t} = v_i + u_{i,t}$ is uncorrelated with $x_{i,t}$, z_i, and $\bar{x}_i$ by construction (under the Mundlak assumptions).

Key Properties: 1. The estimator for β_x from Equation 3 is numerically identical to the FE estimator in the linear case (Mundlak 1978; Wooldridge 2010, chap 10).!! 2. The model allows estimation of β_z, the coefficients on time-invariant variables. 3. A test of $H_0 : \gamma = 0$ provides a robust test for correlation between α_i and $x_{i,t}$, effectively a Hausman-type test comparing FE and RE (Wooldridge 2010).

Handling Unbalanced Panels:

A significant advantage of the Mundlak approach is its straightforward application to unbalanced panels. As shown by Wooldridge (2019, sec 10.7.3), the individual means $\bar{x}_i$ are simply calculated using the available T_i observations for each individual i present in the estimation sample. The estimation of Equation 3 then proceeds using the pooled data. This contrasts with the Chamberlain (1982) approach, which requires conditioning on x_i values from all periods and becomes complex with unbalanced data (Abrevaya 2013). While alternative approaches for unbalanced panels exist, particularly in dynamic contexts (**?**), the use of simple individual means within the static Mundlak framework is well-established and consistent (Wooldridge 2019). The `cre` command implements this approach.

2.2 Nonlinear Models and CRE

The CRE approach becomes particularly valuable for nonlinear models where FE estimation faces challenges. Consider a general nonlinear model where the conditional expectation or relevant latent variable depends on individual effects:

$$E[y_{i,t}|x_{i,t}, z_i, \alpha_i] = g(x_{i,t}\beta_x + z_i\beta_z + \alpha_i)$$

or a latent variable model:

$$y^*_{i,t} = x_{i,t}\beta_x + z_i\beta_z + \alpha_i \tag{4}$$

where $y_{i,t}$ is observed based on $y^*_{i,t}$ (e.g., $y_{i,t} = 1(y^*_{i,t} + u_{i,t} > 0)$ for probit).

Including dummy variables for α_i in nonlinear models generally leads to inconsistent estimates for β_x and β_z as $N \to \infty$ with fixed T, due to the incidental parameter problem (**?**). While consistent FE estimators exist for some specific models (e.g., conditional logit, FE Poisson), they are unavailable for many others (e.g., probit, tobit, ordered models).

Wooldridge (2019, sec 10.7.3) and Wooldridge (2010, chap 15.8) demonstrate that the Mundlak CRE approach extends naturally to these cases. We maintain the assumption from Equation 2 that $\alpha_i = \gamma_0 + \bar{x}_i\gamma + v_i$. The key modeling step is to specify the distribution of α_i (or v_i) conditional on $(\bar{x}_i, z_i)$. A common and convenient assumption is:

$$\alpha_i|\bar{x}_i, z_i \sim N(\gamma_0 + \bar{x}_i\gamma, \sigma^2_v)$$

or simply treat v_i as part of the composite error term after substituting Equation 2 into the model's linear index:

$$y^*_{i,t} = (\beta_0 + \gamma_0) + x_{i,t}\beta_x + z_i\beta_z + \bar{x}_i\gamma + v_i$$

The model can then be estimated by applying the standard pooled estimator (e.g., pooled probit, pooled Tobit) to the augmented specification including $x_{i,t}$, z_i, and $\bar{x}_i$. The parameters β_x, β_z, and γ are consistently estimated under appropriate assumptions for the specific nonlinear model, provided the conditional expectation or density is correctly specified (Wooldridge 2019). Cluster-robust standard errors at the individual level are essential.

This approach provides consistent estimates of the parameters (and subsequently, average partial effects) for both time-varying and time-invariant variables, even when traditional FE methods fail or are unavailable. Wooldridge (2023) utilizes this strategy for estimating treatment effects with staggered adoption in nonlinear settings. The flexibility can be further increased by including interactions between $\bar{x}_i$ and time dummies, or other variables, in the specification (Wooldridge 2019).

2.3 Standard Errors and Hypothesis Testing

Estimating Equation 3 involves using generated regressors ($\bar{x}_i$). While the point estimates for β_x in the linear model are identical to FE, the standard errors are not necessarily the same, even asymptotically, because the inclusion of $\bar{x}_i$ changes the model structure compared to the demeaning process of FE.

For **linear models**, Wooldridge (2010, chap 10.5.3) shows that using the standard RE estimator (GLS) on the augmented equation Equation 3 yields standard errors for β_x that are asymptotically equivalent to the usual FE standard errors. Alternatively, estimating Equation 3 by pooled OLS and using cluster-robust standard errors (clustered at the individual level i) also yields asymptotically valid standard errors, which are equivalent to the clustered FE standard errors. The `cre` command, being a prefix, allows the user to choose the estimation command (e.g., `regress` or `xtreg, re`) and appropriate standard error options (e.g., `vce(cluster id)`).

For **nonlinear models**, the standard approach is to estimate the augmented model using the pooled maximum likelihood estimator (e.g., `probit`, `logit`, `poisson`) and compute standard errors clustered at the individual level (`vce(cluster id)`). This accounts for the within-individual correlation induced by v_i (and potentially $u_{i,t}$ if serially correlated) (Wooldridge 2010, chap 15.8).

A more robust, though computationally intensive, alternative for obtaining standard errors, especially if the distributional assumptions about v_i or $u_{i,t}$ are uncertain, is to **bootstrap** the entire estimation process. This involves resampling individuals (clusters) with replacement, recalculating the $\bar{x}_i$ within each bootstrap sample, estimating the augmented model, and obtaining the distribution of the coefficients across bootstrap replications (Wooldridge 2010, suggests this). This can be achieved in Stata by using the `bootstrap` prefix *before* the `cre` prefix: `bootstrap: cre, abs(idvar) ...`.

2.4 Calculating Average Partial Effects (APEs)

In nonlinear models, the estimated coefficients typically do not directly represent the marginal effect of a covariate on the outcome variable. Instead, Average Partial Effects (APEs) or Marginal Effects at Means (MEMs) are usually reported. A significant practical advantage of the CRE approach implemented via `cre` is the ease of calculating APEs.

Because `cre` explicitly adds the mean variables ($\bar{x}_i$) to the model specification before the core estimation command is executed, the resulting estimation output (`e()`) contains all necessary coefficients (β_x, β_z, γ) and the variance-covariance matrix. Consequently, Stata's powerful `margins` command can be used directly after estimation to compute APEs.

For a time-varying variable $x_{k,it}$ (continuous):

```
cre, abs(idvar) ... : /* estimation command */ ...
margins, dydx(x_k) // Calculates dE[y]/dx_k, averaged over observations
```

For a time-invariant variable $z_{g,i}$ (continuous):

```
cre, abs(idvar) ... : /* estimation command */ ...
margins, dydx(z_g) // Calculates dE[y]/dz_g, averaged over observations
```

The `margins` command correctly accounts for the nonlinear functional form and computes standard errors using the delta method based on the clustered VCE from the estimation step. This seamless integration simplifies the interpretation of results from nonlinear CRE models.

2.5 Interaction Terms

Researchers often want to include interaction terms in their models as well as categorical variable. Methodologogically, adding interactions and categorical variables to the CRE model is straightforward, because there are no changes to the model assuptions. In other words, interactions and categorical variables should be treated as any other variable in the model. From a programatically perspective, `cre` deals with this cases using the following process.

1. All interactions and categorical variables in the main model are expanded (all interactiions appear explicity in the independent variable list).
2. For the case of categorical variables and interactions, temporary variables are created.
3. `cre` then proceeds to create the variable means, for all variables including the temporary ones.
4. As usual, all variable means are added to the model, along with the original variables, and the model is estimated.

Consider an interaction between a time-varying variable $x_{1,it}$ and a time-invariant variable $z_{1,i}$:

```
cre, abs(idvar): /* estimation command */ y x1 x2 c.x1#c.z1 z1 z2 ...
```

Here, `cre` will identify `x1` and `x2` as time-varying and create their means (`_m_x1`, `_m_x2`). It will also identify the *interaction term* `c.x1#c.z1` as potentially time-varying (since x_1 varies over time). It will then compute the individual mean of this interaction term, $\overline{(x_{1,it} \times z_{1,i})}$, and include it as an additional regressor (`_m_x1_z1` or similar).

Similarly, categorical variables can be included in the model:

```
cre, abs(idvar): /* estimation command */ y x1 x2 i.x3 ...
```

`cre` will create means for `x1`, `x2`, and the create means for all levels of `x3` excluding the base level.

Post-estimation commands like `margins` can then be used to compute marginal effects or contrasts involving these interactions:

```
margins, dydx(x1) at(z1=(0 1)) // Effect of x1 at different levels of z1
margins, dydx(x1) dydx(z1) // Check interaction effect interpretation
```

This works because we are interested in estimating the marginal effects of the main variables of interest, after integrating over the unobserved errors, which are assumed fixed. In the Mundlack specification perspective, this implies that created variables, like mean of x, is considered fixed when x changes.

The cre command attempts to create intuitive names for the generated mean variables corresponding to interactions, but complex interactions might result in generic names (_v#) if the generated name exceeds Stata's variable name length limits. The list of generated mean variables is stored in e(m_list) for inspection.

3 cre Command: Implementation in Stata

The cre command is implemented as a Stata prefix command. This means it is placed before a standard Stata estimation command to modify its behavior. cre intercepts the command, identifies the variables and sample, calculates the necessary individual means for time-varying covariates, adds these means to the variable list, and then executes the original estimation command with the augmented specification.

The syntax is:

```
    cre, abs(varlist) [options] : estimation_command depvar [indepvars] [if]
[in] [weight] [, est_options]
```

Required Option:

- abs(varlist): Specifies the variable(s) identifying the groups (individuals) for which means should be calculated. Typically, this is the panel identifier variable (e.g., abs(personid)). Multiple variables can also be specified.[^ The command also allows for the inclusion of multiple variables in the abs() option, allowing for a psude Multi-way Mundlak specification. While this is numerically equivalent to specifying multple fixed effects, the theoretical justification for this is not standard. Nevertheless, it is left as an experimental feature.]

Optional Options:

- prefix(str): Sets the prefix for the generated mean variables. The default is _m. For a variable x, the mean variable will be named _m_x. If multiple variables are specified in abs(), prefixes like _m1_, _m2_ might be used, but the primary use case involves one abs() dimension. Check e(m_list) for generated names.
- hdfe(options): These options are passed directly to the reghdfe command (Correia 2016), which cre uses internally (if available, otherwise it uses egen) to efficiently compute the group means, especially useful for large datasets or complex fixed effects structures within the mean calculation step itself (though the main model only includes the means, not the full FEs). This is mainly for performance tuning.
- dropsingletons: By default, in contrast with reghdfe's typical behavior in absorbing effects, observations belonging to singleton groups (individuals observed only once) are not excluded from the specification. However, one can use dropsingletons to drop these observations from the estimation sample, as its done in reghdfe.

Note: Singletons provide no within-individual variation, so their impact on FE-equivalent estimates is null, but they might influence estimates of time-invariant variables or overall sample size in nonlinear models. Use with caution and understanding of the implications.

- `drop`: If specified, the generated mean variables are dropped from the dataset after the estimation command completes. The default is to keep them for potential inspection or use in post-estimation.

Stored Results:

In addition to the results stored by the `estimation_command`, `cre` adds the following to `e()`: * `e(m_list)`: A list of the names of the generated mean variables added to the model. * `e(abs_vars)`: The variable(s) specified in `abs()`.

Example Usage:

```
* Linear CRE model using regress with clustered SEs
use nlswork, clear
xtset idcode year
cre, abs(idcode): reg ln_wage age tenure race south union, vce(cluster idcode)

* Check generated means
list _m* in 1/10
ereturn list m_list

* Nonlinear CRE model (probit) with APEs
cre, abs(idcode): probit union age tenure i.race i.south, vce(cluster idcode)
margins, dydx(*)

* Using with user-written command (example)
* ssc install ivreg2, replace // If needed
* cre, abs(idcode): ivreg2 lwage (tenure = age), endog(union) ... // Example syntax
```

The command is designed to work with most estimation commands that follow standard Stata syntax. It has been tested with common commands like `regress`, `xtreg`, `probit`, `logit`, `poisson`, `tobit`, `ivregress`, `ivreg2`, etc.

4 Empirical Application

To illustrate the use of `cre` and compare it with alternative estimators, we use the `nlswork.dta` dataset bundled with Stata. This dataset contains panel data on young women from 1968-1988. We will estimate models for the log wage (`ln_wage`) and union membership (`union`). The panel identifier is `idcode`.

```
// Load and setup data
```

```
webuse nlswork, clear
xtset idcode year

// Generate some variables for illustration
gen age_sq = age^2
gen tenure_sq = tenure^2
egen mean_ttl_exp = mean(ttl_exp), by(idcode) // Example time-invariant variable
```

4.1 Linear Model: Log Wage

We estimate a log wage equation using different methods: FE, RE, `xtreg, cre`, and `cre` with `regress`.

```
// Model specification
global linearmod "ln_wage age age_sq tenure tenure_sq i.race i.south"

// 1. Fixed Effects (xtreg, fe)
xtreg $linearmod, fe r // Robust SEs

// 2. Random Effects (xtreg, re)
xtreg $linearmod mean_ttl_exp, re r // Include time-invariant variable

// 3. Stata's built-in CRE (xtreg, cre)
xtreg $linearmod mean_ttl_exp, cre r // Requires StataNow license

// 4. cre prefix with regress
cre, abs(idcode): reg $linearmod mean_ttl_exp, vce(cluster idcode)
ereturn list m_list // See the generated means (_m_age, _m_age_sq, etc.)

// Store estimates for comparison (example using estimates store)
estimates store fe
estimates store re
// estimates store xtreg_cre // If run
estimates store cre_reg
estimates table fe re cre_reg, b(%9.4f) se(%9.4f) stats(N r2_w r2_b r2_o) keep($linearmod _m* m
```

Expected Results Discussion:

- Compare coefficients on time-varying variables (`age`, `age_sq`, `tenure`, `tenure_sq`) between `xtreg, fe` and `cre, ... : reg`. They should be identical. Compare their standard errors (clustered SEs should be very similar).
- Compare `xtreg, re` coefficients with `cre, ... : reg`. They will likely differ, especially if the Hausman test (or the significance of `_m*` vars in the `cre` model) suggests correlation between effects and covariates.

- Highlight that `cre, ... : reg` (and `xtreg, re`) provide estimates for time-invariant `i.race`, `i.south`, and `mean_ttl_exp`, while `xtreg, fe` does not.
- If `xtreg, cre` is available, its results for both time-varying and time-invariant coefficients and SEs should match `cre, ... : reg` when using `vce(cluster idcode)`.
- Discuss the significance of the `_m*` variables in the `cre` output as a test of the RE assumption.

4.2 Nonlinear Model: Union Membership

We estimate a probit model for union membership (`union`). We compare pooled probit, `cre` with `probit`, and potentially `xthybrid` if installed.

```
// Define model variables
global nonlinmod "union age age_sq tenure tenure_sq i.race i.south"

// 1. Pooled Probit (ignores individual effects)
probit $nonlinmod mean_ttl_exp, vce(cluster idcode)
estimates store pooled_p
margins, dydx(*) atmeans // Example APE calculation

// 2. cre prefix with probit
cre, abs(idcode): probit $nonlinmod mean_ttl_exp, vce(cluster idcode)
estimates store cre_p
margins, dydx(*) // Calculate APEs

// 3. (Optional) xthybrid comparison
* ssc install xthybrid, replace // If needed
* xthybrid $nonlinmod mean_ttl_exp, family(binomial) link(probit) vce(cluster idcode) cluster(i
* estimates store xth_p
* margins, dydx(*) // Check if margins works easily after xthybrid

// Compare APEs (example)
estimates restore cre_p
margins, dydx(age tenure i.race i.south mean_ttl_exp) post
matrix m_cre = r(table)'

estimates restore pooled_p
margins, dydx(age tenure i.race i.south mean_ttl_exp) post
matrix m_pooled = r(table)'

* Display comparison table (manually or using tools)
matrix list m_pooled
matrix list m_cre
// Potentially compare xthybrid APEs if run
```

Expected Results Discussion: * Compare the coefficients and especially the APEs from the pooled probit and the `cre` probit. Highlight expected differences due to controlling for unobserved heterogeneity via CRE. * Demonstrate the ease of obtaining APEs using `margins` after `cre, ... : probit`. * Show that `cre` allows estimating the effect of time-invariant variables (`i.race`, `i.south`, `mean_ttl_exp`) in the probit model while accounting for correlation with individual effects. * If `xthybrid` is used, compare its results (coefficients/APEs) and ease of use (especially with `margins`) to the `cre` approach. Note any differences in sample size or estimates.

This empirical section demonstrates the flexibility of `cre` for both linear and nonlinear models, its ability to handle time-invariant variables, its compatibility with post-estimation commands, and provides a basis for comparison with existing methods.

5 Monte Carlo Simulations

To assess the performance of the `cre` command, particularly in the nonlinear context where FE is often problematic, we conducted a Monte Carlo simulation study. We consider a panel data generating process (DGP) with one unobserved individual fixed effect (α_i) correlated with the explanatory variables. We simulate data for $N = 1000$ individuals over $T = 5$ periods (allowing for some attrition to create an unbalanced panel).

The DGP is as follows:

```
// Simulation Setup (Conceptual Code)
clear
set seed 12345
local N = 1000
local T = 5
set obs `N'
gen id1 = _n // Individual ID

// Generate correlated fixed effect
gen c1 = rnormal()

// Generate correlated time-varying regressors
gen x1_base = rnormal() + 0.5 * c1
gen x2_base = rnormal() - 0.5 * c1

// Expand to panel
expand `T'
bysort id1: gen time = _n
xtset id1 time

// Add noise and time variation
```

```
gen x1 = x1_base + rnormal()*0.5
gen x2 = x2_base + rnormal()*0.5

// Latent outcome variable (example: linear index)
// True coefficients: beta0=0, beta_x1=1, beta_x2=0.5, effect_alpha=1
gen y_star = 1 * x1 + 0.5 * x2 + 1 * c1 + rnormal() // Added overall error

// Simulate Unbalanced Panel (randomly drop ~20% person-years)
gen drop_obs = runiform() < 0.2
drop if drop_obs & time > 1 // Don't drop first obs

xtset id1 time // Reset panel structure after dropping
```

We focus on the performance of `cre` for four nonlinear models derived from `y_star`: probit, fractional probit, tobit, and poisson.

```
// Generate Observed Outcomes (Conceptual Code)
// Probit
gen y_probit = (y_star > 0)
// Fractional Probit
gen y_fprobit = normal(y_star/sd(y_star)) // Standardize index for 0-1 range
replace y_fprobit = 0 if y_fprobit < 0
replace y_fprobit = 1 if y_fprobit > 1
// Tobit (left-censored at 0)
gen y_tobit = max(0, y_star)
// Poisson
gen y_poisson = rpoisson(exp(y_star / 4)) // Rescale index to avoid huge counts
```

For each model type, we estimate three specifications over 1000 Monte Carlo replications: 1. **Unfeasible Benchmark:** Model estimated including the true fixed effect `c1` as a regressor. 2. **Pooled Estimator:** Model estimated ignoring `c1` (and using cluster-robust SEs). 3. **CRE Estimator:** Model estimated using `cre, abs(id1): ...` including `x1`, `x2` but not `c1`.

Example estimation commands within the simulation loop:

```
// Probit Example inside loop
// 1. Benchmark
probit y_probit x1 x2 c1, vce(cluster id1)
// 2. Pooled
probit y_probit x1 x2, vce(cluster id1)
// 3. CRE
cre, abs(id1): probit y_probit x1 x2, vce(cluster id1)
```

We compare the distribution of estimated coefficients (or APEs for probit/fractional) for `x1` and `x2` across the methods, focusing on bias and Mean Absolute Error (MAE) relative to the unfeasible benchmark average. We use the `parallel` command (Vega Yon and Quistorff 2019) for efficiency. *(Self-note: Ensure the simulation code file provided is complete and functional).*

The results of the simulation are presented in Figure 1 and Table 1. Figure 1 shows the densities of the estimated coefficients (or APEs) for the key parameters across simulations, while Table 1 summarizes the bias and MAE.

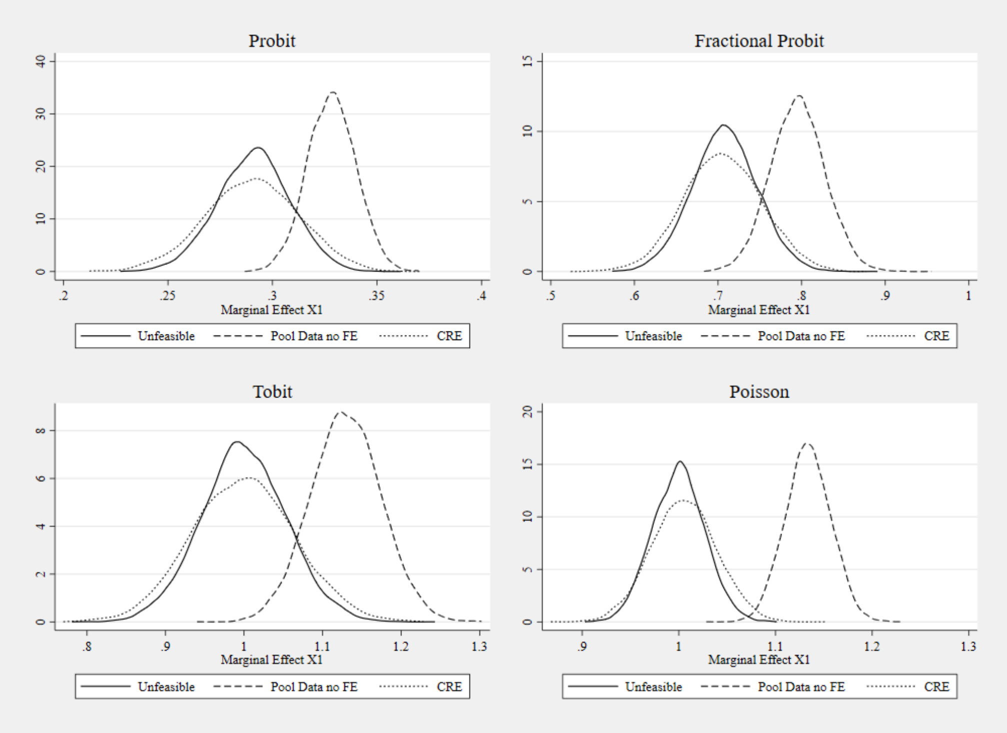

Figure 1: *Estimated marginal effects/Coefficient densities for non-linear models*

As expected, the unfeasible estimator, controlling directly for the unobserved effect `c1`, provides the benchmark estimates. Since the unobserved effect `c1` is correlated with `x1` and `x2` by construction, the pooled estimators (which ignore `c1`) exhibit significant bias, as seen in both the density plots and the summary table.

In contrast, the CRE approach, implemented using the `cre` prefix, yields estimates whose distributions are centered close to the benchmark estimates, indicating negligible bias. While the CRE estimates show slightly higher variance compared to the unfeasible benchmark (as reflected in slightly larger MAE in Table 1), they effectively mitigate the bias caused by the omitted correlated fixed effect. This demonstrates the utility of the CRE method for obtaining consistent estimates in nonlinear panel models where FE is not viable and RE assumptions are violated.

Table 1: *Bias and MAE for the estimated marginal effects/Coefficients for non-linear models*

{< include simulation/table1.txt >}

6 Conclusion

This paper introduced `cre`, a versatile Stata prefix command designed to simplify the estimation of Correlated Random Effects (CRE) models based on the Mundlak (1978) specification. The CRE approach provides a valuable bridge between standard Fixed Effects and Random Effects models, offering several advantages: it allows for the estimation of time-invariant variable effects (unlike FE) while providing consistent estimates for time-varying coefficients even when the strict RE exogeneity assumption fails (matching FE estimates in linear models).

The primary contribution of the `cre` command lies in its **flexibility and ease of use**. As a prefix command, it can be applied to a wide array of Stata's linear and nonlinear estimation commands, including user-written ones. It automatically handles the generation of individual means for time-varying covariates, supports both balanced and unbalanced panels using established methods (Wooldridge 2019), and integrates seamlessly with factor variables and post-estimation tools like `margins` for calculating Average Partial Effects, which is particularly crucial for interpreting nonlinear models.

We demonstrated through empirical examples that `cre` replicates the behavior of specialized commands like `xtreg, cre` in the linear case while extending functionality to other estimators. For nonlinear models, we showed how `cre` provides a practical way to obtain consistent estimates and APEs, addressing the limitations of FE and the potentially strong assumptions of RE. Monte Carlo simulations further confirmed that the CRE approach implemented by `cre` effectively reduces bias in nonlinear models with correlated unobserved effects.

While `cre` offers a significant simplification for estimating static CRE models, it currently **does not address dynamic panel models**. The inclusion of lagged dependent variables introduces further econometric challenges (e.g., the initial conditions problem) that require specialized estimators beyond the scope of this command.

In summary, the `cre` command provides applied researchers with a user-friendly and powerful tool for leveraging the benefits of the Correlated Random Effects approach in Stata, making it easier to estimate models that account for unobserved heterogeneity while retaining the ability to analyze the effects of time-invariant characteristics, especially in nonlinear settings.

7 Acknowledgments

Thanks to Aashima Sinha for her help in the preparation and providing feedback for this paper, and Enrique Pinzon for his encouragement on pushing this project forward. I am also grateful to the Editor and an anonymous referee for their constructive comments on a previous version, which significantly improved the focus and clarity of this paper.

8 References

Abrevaya, J. 2013. The projection approach for unbalanced panel data. *The Econometrics Journal* 16(2): 161–178. http://dx.doi.org/10.1111/j.1368-423X.2012.00389.x.

Chamberlain, G. 1982. Multivariate regression models for panel data. *Journal of econometrics* 18(1): 5–46. https://doi.org/10.1016/0304-4076(82)90094-X.

Correia, S. 2016. A Feasible Estimator for Linear Models with Multi-Way Fixed Effects. *Unpublished Manuscript* .

Mundlak, Y. 1978. On the pooling of time series and cross section data. *Econometrica: journal of the Econometric Society* 46(1): 69–85. https://doi.org/10.2307/1913646.

Perales, F. 2013. MUNDLAK: Stata module to estimate random-effects regressions adding group-means of independent variables to the model. Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s457601.html.

Schunck, R., and F. Perales. 2017. Within- and Between-cluster Effects in Generalized Linear Mixed Models: A Discussion of Approaches and the Xthybrid command. *The Stata Journal* 17(1): 89–115. https://doi.org/10.1177/1536867X1701700106.

Vega Yon, G. G., and B. Quistorff. 2019. parallel: A command for parallel computing. *The Stata Journal* 19(3): 667–684. https://doi.org/10.1177/1536867X19874242.

Wooldridge, J. M. 2010. *Econometric analysis of cross section and panel data*. MIT press.

———. 2019. Correlated random effects models with unbalanced panels. *Journal of Econometrics* 211(1): 137–150.

———. 2023. Simple approaches to nonlinear difference-in-differences with panel data. *The Econometrics Journal* 26(3): C31–C66. https://doi.org/10.1093/ectj/utad016.

About the authors

Fernando Rios-Avila is an applied econometrician with passion for econometrics and programming. His research interests include applied econometrics, labor economics, and poverty and inequality. He has contributed many commands to Statistical Software Components and written articles for the Stata Journal.