



*The World's Largest Open Access Agricultural & Applied Economics Digital Library*

**This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.**

**Help ensure our sustainability.**

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

[aesearch@umn.edu](mailto:aesearch@umn.edu)

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

*No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.*

The Stata Journal (2016)  
16, Number 4, pp. 826–836

# A generalized regression-adjustment estimator for average treatment effects from panel data

David M. Drukker  
StataCorp  
College Station, TX  
ddrukker@stata.com

**Abstract.** I illustrate that the simple regression-adjustment estimator is inconsistent for the average treatment effect when the random effects affecting treatment assignment are correlated with the random effects that affect the potential outcomes. I present a simple parametric estimator that is consistent in this case.

**Keywords:** st0456, panel data, longitudinal data, random effects, causal inference, treatment effect, regression adjustment, endogenous treatment, gsem

## 1 Introduction

When one correlates the random effects that affect treatment assignment with the random effects that affect the treatment-specific potential outcomes, the simple regression estimator (SRA) produces inconsistent results. Many do not appreciate this problem's importance, because many view random effects only as causing problems for estimating the standard errors (SEs). I discuss this issue and present a generalized regression-adjustment (GRA) estimator consistent for the average treatment effect (ATE) at time  $t$  from panel data. I also discuss consistent SEs for this estimator and Monte Carlo simulations showing that the GRA estimator performs well and that an SRA produces inconsistent results.

## 2 The model

A model of the treatment-assignment process and separate models for the outcomes one would obtain if assigned to each potential treatment level are essential to the potential-outcome model used to construct treatment-effect estimators. See [Imbens \(2004\)](#), [Imbens and Wooldridge \(2009\)](#), [Wooldridge \(2010\)](#), and [Imbens and Rubin \(2015\)](#) for introductions to these models. Panel-data models invariably allow for the presence of individual-level effects; for background, see [Baltagi \(2013\)](#), [Cameron and Trivedi \(2005\)](#), and [Wooldridge \(2010\)](#). I present a GRA estimator for the average treatment effect at time  $t$  ( $ATE_t$ ) based on potential-outcome models that incorporate individual-level effects. I specify that the individual-level effects are random, that is, independent of the covariates and the idiosyncratic errors in the models. A random-effects probit model for treatment assignment is

$$\tau_{it} = \begin{cases} 1 & \text{if } \mathbf{z}_{it}\boldsymbol{\gamma}' + \alpha_i + \xi_{it} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where  $\mathbf{z}_{it}$  are the treatment covariates,  $\boldsymbol{\gamma}$  are the coefficients on  $\mathbf{z}_{it}$ ,  $\alpha_i$  is the individual-level random effect affecting treatment, and  $\xi_{it}$  is the idiosyncratic shock affecting treatment. In the random-effects probit model used here,  $\alpha_i$  is a zero-mean normal with standard deviation (SD)  $\sigma_\alpha$ , and  $\xi_{it}$  is a standard normal.

Specifying linear models for the two potential outcomes yields

$$y_{0,it} = \mathbf{x}_{it}\boldsymbol{\beta}_0 + \eta_i + \epsilon_{0,it} \quad (2)$$

$$y_{1,it} = \mathbf{x}_{it}\boldsymbol{\beta}_1 + \eta_i + \epsilon_{1,it} \quad (3)$$

where  $y_{0,it}$  is the potential outcome that person  $i$  would obtain at time  $t$  if assigned to treatment 0,  $y_{1,it}$  is the potential outcome that person  $i$  would obtain at time  $t$  if assigned to treatment 1,  $\mathbf{x}_{it}$  are outcome covariates,  $\boldsymbol{\beta}_0$  are the coefficients on  $\mathbf{x}_{it}$  when  $i$  is assigned to treatment 0,  $\boldsymbol{\beta}_1$  are the coefficients on  $\mathbf{x}_{it}$  when  $i$  is assigned to treatment 1,  $\eta_i$  is the individual-level random effect, and  $\epsilon_{it}$  is the time-varying idiosyncratic shock.

The  $\text{ATE}_t$  is the difference between the means of these potential outcomes at time  $t$ ,

$$\text{ATE}_t = E(y_{1,it} - y_{0,it}) = E(y_{1,it}) - E(y_{0,it})$$

where, in a slight abuse of notation, the expectation is over the individuals  $i$  and time  $t$  is held fixed.

The specification in (1)–(3) defines an endogenous treatment effect model when the treatment-assignment random effect  $\alpha_i$  and the potential-outcome random effect  $\eta_i$  are correlated, which is probably the case.

Three features of the above model stand out:

1. An SRA estimator that ignores the endogeneity caused by the correlation between the treatment-assignment random effect and the potential-outcome random effect will be inconsistent. This problem cannot be fixed by using a robust estimator of the SEs, because the point estimator is inconsistent.

The correlation between the treatment-assignment random effect and the potential-outcome random effect creates correlation between the unobservables that affect treatment assignment and the unobservables that affect the potential outcomes, which violates the conditional mean independence (CMI) assumption required for the consistency of simple regression-adjustment (RA) estimators. See Imbens and Wooldridge (2009) and Wooldridge (2010) for discussions of why RA estimators require CMI and why correlations between unobservables that affect treatment assignment and potential outcomes violate CMI.

Another way of explaining why an SRA is inconsistent notes that the correlation between the treatment-assignment random effect and the potential-outcome random effect creates an endogenous sample-selection problem. An SRA that ignores

this endogeneity produces inconsistent estimates of the regression parameters for the potential outcomes, causing the difference in the predicted means to be inconsistent for the  $ATE_t$ .

For example, if  $\alpha_i$  and  $\eta_i$  are positively correlated, individuals with larger random effects will be more likely observed in the  $y_1$  group and have above-average  $y_1$  values, causing the SRA estimator to overestimate the true ATE.

2. One can use maximum likelihood (ML) estimators for the parameters of generalized structural-equation models (GSEM) to construct a consistent GRA estimator at the cost of parametric assumptions.

GSEMs are nonlinear triangular models that share random effects and random coefficients; see [Rabe-Hesketh and Skrondal \(2012\)](#) for an introduction to these models. Fully parametric ML estimators are relatively simple estimators for the parameters; see [Rabe-Hesketh and Skrondal \(2012\)](#). The unobserved random errors are assumed to have joint normal distributions.

A GRA estimator is the difference in predicted means using ML estimates of GSEM parameters.

3. `gsem` can estimate the parameters of GSEMs, and `margins` can estimate the ATE using the parameters estimated by `gsem`. This estimation framework produces consistent point estimates and a consistent estimator for the variance–covariance of the estimator. ML jointly estimates the random-effects parameters and the conditional mean parameters. `margins` accounts for the two-step estimation problem using the standard method discussed by [Wooldridge \(2010, chaps. 12 and 13\)](#) and [Cameron and Trivedi \(2005, chap. 6.6\)](#).

In a simple random-effects model, one assumes no functional forms for the distributions of  $\eta_i$ ,  $\epsilon_{0,it}$ , or  $\epsilon_{1,it}$ . The model requires more structure because the treatment-assignment random effect is correlated with the potential-outcome random effects. The estimators discussed here assume that these random variables are jointly normally distributed with  $\alpha_i$  and  $\xi_{it}$ . In section 5, I discuss extensions that drop the assumption of joint normality.

### 3 RA estimators

RA estimators for the  $ATE_t$  have the following structure:

- They estimate  $\beta_1$  and  $\beta_0$  using regression estimators  $\hat{\beta}_1$  and  $\hat{\beta}_0$ .
- They estimate the mean of the treatment level 1 potential outcome at time  $t$   $E(y_{1,it})$  by the mean of the predicted values, using  $\hat{\beta}_1$  for all individuals at time  $t$ .
- They estimate the mean of the treatment level 0 potential outcome at time  $t$   $E(y_{0,it})$  by the mean of the predicted values, using  $\hat{\beta}_0$  for all individuals at time  $t$ .
- They estimate the  $ATE_t$  by the difference in these mean predictions.

### 3.1 An inconsistent SRA estimator

An inconsistent SRA estimator for the  $ATE_t$  that ignores the endogeneity uses generalized least-squares estimators for each  $\hat{\beta}_{\bar{\tau}}$ , using only the observations for which  $\tau_{it} = \bar{\tau}$ . Each of these generalized least-squares estimators is inconsistent, because each ignores the endogenous sample-selection problem. Each of the mean predictions is inconsistent, because each uses inconsistent estimators for its respective  $\beta_{\bar{\tau}}$  parameters. The resulting difference in mean predictions is inconsistent, because both mean predictions are inconsistent.

Below is an example of some simulated data.

```
. use xttesim1
. quietly xtreg y x1 x2 if d==1, re
. predict double yh1, xb
. quietly xtreg y x1 x2 if d==0, re
. predict double yh0, xb
. mean yh1 yh0, over(t) coeflegend
Mean estimation      Number of obs   =      6,000
      1: t = 1
      2: t = 2
      3: t = 3
```

	Over	Mean	Legend
yh1	1	1.39922	_b[yh1:1]
	2	1.400209	_b[yh1:2]
	3	1.382231	_b[yh1:3]
yh0	1	.3668965	_b[yh0:1]
	2	.3844787	_b[yh0:2]
	3	.3635778	_b[yh0:3]

```
. display "ATE at time 1 is " _b[yh1:1] - _b[yh0:1]
ATE at time 1 is 1.0323234
. display "ATE at time 2 is " _b[yh1:2] - _b[yh0:2]
ATE at time 2 is 1.0157304
. display "ATE at time 3 is " _b[yh1:3] - _b[yh0:3]
ATE at time 3 is 1.0186535
```

Because the data are simulated using the code in appendix A, I know the true values. The reported estimates of 1.40, 1.40, and 1.38 are not close to the true values of  $E(y_{1,it})$ , which equal 1.2 for each  $t$ . The reported estimates of 0.37, 0.38, and 0.36 are not close to the true values of  $E(y_{0,it})$ , which equal 0.8 for each  $t$ . The reported estimates of 1.03, 1.02, and 1.02 are not close to the true values of the  $ATE_t$ , which equal 0.4 for each  $t$ .

To support the assertion that the estimates of 1.0 are not close to the true value of 0.4, I present the results from an estimator that uses quasi-ML estimates for the coefficients in each potential-outcome equation. This estimator is implemented in `gsem`.

`gsem` allows me to use `margins` to estimate the means of the potential outcomes at each  $t$  and to use `nlcom` to estimate the differences of the estimated means.

```
. generate y1_obs = y if d==1
(1,910 missing values generated)
. generate y0_obs = y if d==0
(4,090 missing values generated)
. quietly gsem (y1_obs <- x1 x2 U1[id]) (y0_obs <- x1 x2 U0[id]),
> covstructure(U1[id] U0[id], diagonal) vce(robust)
. margins, predict(outcome(y1_obs) marginal) predict(outcome(y0_obs) marginal)
> vce(unconditional) post over(t) coeflegend

Predictive margins                                Number of obs      =        6,000
over               : t
1._predict         : Marginal predicted mean (y1_obs), predict(outcome(y1_obs) marginal)
2._predict         : Marginal predicted mean (y0_obs), predict(outcome(y0_obs) marginal)
                    (Std. Err. adjusted for 2,000 clusters in id)
```

	Margin	Legend
._predict#t		
1 1	1.399633	_b[1bn._predict#1bn.t]
1 2	1.400617	_b[1bn._predict#2.t]
1 3	1.382645	_b[1bn._predict#3.t]
2 1	.3681741	_b[2._predict#1bn.t]
2 2	.3857747	_b[2._predict#2.t]
2 3	.3648502	_b[2._predict#3.t]

```
. nlcom ( _b[1bn._predict#1bn.t] - _b[2._predict#1bn.t] )
> ( _b[1bn._predict#2.t] - _b[2._predict#2.t] )
> ( _b[1bn._predict#3.t] - _b[2._predict#3.t] ),
> noheader
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_nl_1	1.031459	.0422238	24.43	0.000	.9487014	1.114216
_nl_2	1.014843	.0424152	23.93	0.000	.9317102	1.097975
_nl_3	1.017794	.0423458	24.04	0.000	.9347982	1.100791

In the code above, I generate `y1_obs`, which contains either the observed value of  $y_{1,it}$  or a missing value. (`y0_obs` has the same structure.) These variables are the outcome variables in the subsequent `gsem` command. (See [SEM] **example 45g** for more about how `gsem` handles missing values.) I then use `margins`, which uses the point estimates produced by `gsem` to estimate the means of each predicted potential outcome using all the individuals for each time period. Finally, I use `nlcom` to estimate the differences in the predicted potential outcome for each time period. The SEs, which account for first-stage estimation error, support the previous assertion that the estimates of 1.0 are not close to the true values of 0.4.

### 3.2 A consistent GRA estimator

A consistent GRA estimator for the  $ATE_t$  uses ML estimators for  $\hat{\beta}_0$  and  $\hat{\beta}_1$  based on the GSEM model, which accounts for the endogenous sample-selection problem. Each mean prediction is consistent because each uses consistent estimators for its respective  $\beta_\tau$  parameters. The resulting difference in mean predictions is consistent because both mean predictions are consistent.

The code below implements this GRA estimator.

```
. quietly gsem (y1_obs <- x1 x2 U1[id]) (y0_obs <- x1 x2 U1[id])
> (d <- x1 x2 x3 U[id]@1, probit), cov(U1[id]*U[id]) vce(robust)

. margins, predict(outcome(y1_obs) marginal) predict(outcome(y0_obs) marginal)
> vce(unconditional) post over(t) coeflegend

Predictive margins                                Number of obs      =        6,000
over               : t
1._predict         : Marginal predicted mean (y1_obs), predict(outcome(y1_obs) marginal)
2._predict         : Marginal predicted mean (y0_obs), predict(outcome(y0_obs) marginal)
                    (Std. Err. adjusted for 2,000 clusters in id)
```

	Margin	Legend
._predict#t		
1 1	1.227004	_b[1bn._predict#1bn.t]
1 2	1.228521	_b[1bn._predict#2.t]
1 3	1.20972	_b[1bn._predict#3.t]
2 1	.8174044	_b[2._predict#1bn.t]
2 2	.8352209	_b[2._predict#2.t]
2 3	.8130332	_b[2._predict#3.t]

```
. nlcom ( _b[1bn._predict#1bn.t] - _b[2._predict#1bn.t] )
> ( _b[1bn._predict#2.t] - _b[2._predict#2.t] )
> ( _b[1bn._predict#3.t] - _b[2._predict#3.t] ),
> noheader
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
_nl_1	.4096	.0450441	9.09	0.000	.3213153	.4978848
_nl_2	.3932996	.0451034	8.72	0.000	.3048985	.4817007
_nl_3	.3966868	.0452209	8.77	0.000	.3080555	.4853182

Only the `gsem` command in the GRA code differs from the `gsem` version of the code that implemented the SRA in the previous subsection. The `margins` and `nlcom` commands are the same, because both the RA aspects of the SRA and the GRA estimators are the same.

The `gsem` command in the GRA code accounts for the endogeneity by specifying the triangular structure of the potential-outcome model and estimating the correlation between the treatment-assignment random effect and the potential-outcome random effects.

The estimated  $ATE_t$  for each  $t$  is close to the true value of 0.4.

## 4 A Monte Carlo simulation

### 4.1 A DGP for the potential-outcome model

The data-generating process (DGP) for the simulations is a specific case of the potential-outcome model specified in section 2.

I chose a DGP with two important features:

1. I include a variable in  $\mathbf{z}_{it}$  that is not included in  $\mathbf{x}_{it}$ . Like many endogenous selection models, one can identify the parameters using the nonlinearity in the functional form, but this identification can be weak. I abstract from this issue by including  $z_{3it}$  in (1) but not in (1) or (3).
2. Conditional on  $\mathbf{z}_{it}\boldsymbol{\gamma}'$  and  $\mathbf{x}_{it}\boldsymbol{\beta}'$ , the variances and covariance of  $\alpha_i$  and  $\eta_i$  determine the extent to which the SRA estimator is inconsistent.

For clarity, I chose the parameter values so that the true ATE is 0.4 in each  $t$ , but the SRA estimator produces estimates around 1.0.

See appendix A for details.

### 4.2 A Monte Carlo

In all 2,000 repetitions, I performed the following tasks:

1. I drew a balanced panel with 3 time periods for 1,000 individuals.
2. I computed the SRA estimator from section 3.1 and stored the results.
3. I computed the GRA estimator from section 3.2, performed Wald tests of the null hypotheses that the ATE in each  $t$  is 0.4, and stored the results.



The table below summarizes the results.

Table 1. Simulation results: True ATE is 0.4 for each  $t$

Parameter	Estimator	Mean	SD	SE	Rejection rate
ATE <sub>1</sub>	SRA	1.02	0.058		
ATE <sub>1</sub>	GRA	0.40	0.062	0.062	0.054
ATE <sub>2</sub>	SRA	1.02	0.058		
ATE <sub>2</sub>	GRA	0.40	0.062	0.062	0.050
ATE <sub>3</sub>	SRA	1.02	0.058		
ATE <sub>3</sub>	GRA	0.40	0.062	0.062	0.055

For each time period, the SRA estimator is tightly distributed around the wrong value of 1.02. There is no need to compute SEs or rejection rates for the SRA estimator, because the estimates are so far from the true value.

In contrast, for each time period, the GRA estimator is tightly distributed around the true value of 0.40. Moreover, the SD of the GRA estimates is close to the mean of the estimated SEs, and the rejection rate is close to 0.05.

## 5 Conclusions and extensions

I presented a GRA estimator for the ATE in each period from panel data. If one correlates the random effects that affect treatment with the random effects that affect the potential outcomes, the SRA estimator is inconsistent. However, a fully parametric GRA estimator is consistent and performed well in a Monte Carlo simulation.

The GRA estimator can estimate the ATE on the treated or untreated by restricting the sample over which `margins` computes the means.

It would be simple to extend the parametric ML estimator to allow for correlated random effects that a Mundlak or Chamberlain device can model; see [Wooldridge \(2010, chap. 11\)](#).

A control function estimator that does not assume that the treatment-level random effect and the potential-outcome random effect are jointly normal warrants further study.

One could extend either of the previous two extensions to allow for binary or exponential-conditional-mean potential outcomes.

## 6 Acknowledgment

I would like to thank an anonymous referee for useful comments.

## 7 References

- Baltagi, B. H. 2013. *Econometric Analysis of Panel Data*. 5th ed. New York: Wiley.
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Imbens, G. W. 2004. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and Statistics* 86: 4–29.
- Imbens, G. W., and D. B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. New York: Cambridge University Press.
- Imbens, G. W., and J. M. Wooldridge. 2009. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature* 47: 5–86.
- Rabe-Hesketh, S., and A. Skrondal. 2012. *Multilevel and Longitudinal Modeling Using Stata*. 3rd ed. College Station, TX: Stata Press.
- Wooldridge, J. M. 2010. *Econometric Analysis of Cross Section and Panel Data*. 2nd ed. Cambridge, MA: MIT Press.

### About the author

David M. Drukker is the executive director of econometrics at StataCorp.

## A Appendix: Code for the DGP

```

program mkdata
    version 14
    drop _all
    local N = `1'
    local T = `2'
    set obs `N'
    generate id = _n
    matrix C = (1, .7 .7, 1)
    drawnorm ut ud, cov(C)
    expand `T'

    sort id
    by id: generate t = _n
    xtset id t

    generate c = rnormal()
    generate x1 = rnormal() + .5*c + 1
    generate x2 = rnormal() + .5*c + 1
    generate x3 = rnormal() + .5*c + 1

    generate e0 = rnormal()
    generate e1 = rnormal()
    generate e2 = rnormal()

    local b00 = .1
    local b01 = .2
    local b02 = .5
    local b10 = .1

```

```

        local b11 = .5
        local b12 = .6

        generate d = (.1 + .1*x1 + .2*x2 + .3*x3 + ut + e2 > 0)
        generate y0 = `b00' + `b01'*x1 + `b02'*x2 + ud + e0
        scalar Ey0 = `b00' + `b01'      + `b02'
        generate y1 = `b10' + `b11'*x1 + `b12'*x2 + ud + e1
        scalar Ey1 = `b10' + `b11'      + `b12'
        scalar ATE = Ey1 - Ey0
        generate y = d*y1 + (1-d)*y0
    end

```

## B Appendix: Code for the simulations

```

// Do simulations reported in xttesem article
version 14
clear all
set seed 12345671
local N 1000
local T 3
local R 2000
use xttesim1
quietly generate y1_obs = y if d==1
quietly generate y0_obs = y if d==0
quietly gsem (y1_obs <- x1 x2 U1[id])          ///
              (y0_obs <- x1 x2 U1[id])          ///
              (d <- x1 x2 x3 U[id]@1, probit),  ///
              cov(U1[id]*U[id]) vce(robust) iter(0)
mat bs = e(b)
mat b0 = (.5, .6, 1, .1, .2, .5, 1, .1, .1, .2, .3, 1, .1, 1, 1, .75, 1, 1)
forvalues j = 1/18 {
    matrix bs[1,`j'] = b0[1,`j']
}
postfile sim sra_1 sra_2 sra_3                ///
          gra_1 gra_1se gra_1r                ///
          gra_2 gra_2se gra_2r                ///
          gra_3 gra_3se gra_3r                ///
          using xttesemsim, replace
forvalues i = 1/`R' {
    quietly mkdata `N' `T'
    // display "Ey1 = " Ey1 " Ey0 = " Ey0 " ATE = " ATE
    quietly xtreg y x1 x2 if d==1, re
    predict double yh1, xb
    quietly xtreg y x1 x2 if d==0, re
    predict double yh0, xb
    quietly mean yh1 yh0, over(t) coeflegend
    local sra_1 = _b[yh1:1] - _b[yh0:1]
    local sra_2 = _b[yh1:2] - _b[yh0:2]
    local sra_3 = _b[yh1:3] - _b[yh0:3]
    quietly generate y1_obs = y if d==1
    quietly generate y0_obs = y if d==0
    quietly gsem (y1_obs <- x1 x2 U1[id])          ///
                  (y0_obs <- x1 x2 U1[id])          ///
                  (d <- x1 x2 x3 U[id]@1, probit),  ///
                  cov(U1[id]*U[id]) vce(robust) from(bs)
}

```

```

quietly margins, predict(outcome(y1_obs) )           ///
predict(outcome(y0_obs) )                           ///
vce(unconditional) post over(t)                     ///
coeflegend
quietly nlcom ( _b[1bn._predict#1bn.t] - _b[2._predict#1bn.t] ) ///
( _b[1bn._predict#2.t] - _b[2._predict#2.t] )        ///
( _b[1bn._predict#3.t] - _b[2._predict#3.t] ),        ///
noheader post
local gra_1 = _b[_nl_1]
local gra_1se = _se[_nl_1]
quietly test _b[_nl_1] = .4
local gra_1r = (r(p)<.05)
local gra_2 = _b[_nl_2]
local gra_2se = _se[_nl_2]
quietly test _b[_nl_2] = .4
local gra_2r = (r(p)<.05)
local gra_3 = _b[_nl_3]
local gra_3se = _se[_nl_3]
quietly test _b[_nl_3] = .4
local gra_3r = (r(p)<.05)
post sim (`sra_1') (`sra_2') (`sra_3')              ///
(`gra_1') (`gra_1se') (`gra_1r')                    ///
(`gra_2') (`gra_2se') (`gra_2r')                    ///
(`gra_3') (`gra_3se') (`gra_3r')
if floor(`i'/50) == `i'/50 {
    display ".      `i'"
}
else {
    display _c ". "
}
}
postclose sim
use xttesemsim, clear
summarize, sep(3)

```