# New Estimates for *Propensity Score Analysis* Monte Carlo Simulations

Paul E. Johnson <pauljohn@ku.edu>
Center for Research Methods and Data Analysis
University of Kansas
Lawrence, Kansas 66045

January 15, 2016

A new set of estimates for the Monte Carlo exercise reported in Guo and Fraser (2015, pp. 347-357) is presented. After correcting a minor defect in the Stata code that was used in the original exercise, substantial changes are evident. The focal point of concern is Stata's `corr2data` function. `corr2data` should not be used when draws from a multivariate normal distribution are needed.

In their widely read text *Propensity Score Analysis: Statistical Methods and Applications* (Guo & Fraser, 2015; henceforth, *PSM)*, professors Shenyang Guo and Mark W. Fraser present a comprehensive, readable survey of methods for statistical estimation of treatment effects in quasi-experimental research designs. The book includes a set of Monte Carlo simulation estimates which indicate that some estimation procedures are biased (on average, far from the correct values) and lower in variance than others.

In this note, we show that a minor flaw in the Stata (StataCorp, 2015) code used in *PSM* can be corrected to yield estimates that are substantially different. Stata's `corr2data` function has unanticipated effects and leads to questionable results that were reported in Table 11.2 of *PSM*(2015, p. 452). This note does not offer a comprehensive re-analysis of all of the proposed methods, but it does emphasize some changes in the interpretation that must necessarily follow from software revision. Professors Guo and Fraser wrote that, "We encourage readers to replicate the study and to use our syntax as a baseline to generate more settings or to compute additional models" (2015, p. 349). It is reasonable to expect that researchers will want to continue along this line. They should be warned about the problems caused by the reliance on `corr2data`.

The report proceeds as follows. First, a summary of the data generating process envisaged by Guo and Fraser is outlined. Second, the Stata implementation is reviewed. Third, revised estimates from Stata are discussed.

## The Original Model

The simulation model is described in Chapter 11 of the second edition of *Propensity Score Analysis* (Guo & Fraser 2015; Chapter 8 of the first edition, Guo & Fraser, 2009). "The Stata syntax generates Setting 1 using the following specifications:

$$\begin{aligned} Y &= 100 + .5x_1 + .2x_2 - .05x_3 + .5W + u \qquad (1)\\ W* &= .5Z + 1x_3 + v, \end{aligned}$$

where $x_1$, $x_2$, $x_3$, $Z$, and $u$ are random variables, normally distributed with a mean vector of (3 2 10 5 0), a standard deviation vector (.5 .6 9.5 2 1) and the following symmetric correlation matrix:

$$r(x_1, x_2, x_3, Z, u) = \begin{bmatrix} 1 & .2 & .3 & 0 & 0 \\ .2 & 1 & 0 & 0 & 0 \\ .3 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & .4 \\ 0 & 0 & 0 & .4 & 1 \end{bmatrix}. \qquad (2)$$

In addition, $v$ is a random variable that is normally distributed with mean zero and variance 1; $W = 1$, if $W* >$ Median($W*$), and $W = 0$ otherwise" (2015, p. 350). The focal point in this exercise is ability of various estimators to recover a causal treatment effect, which is set at 0.5 (the coefficient for the variable $W$). The multivariate draw of $x_1, x_2, x_3, Z, u$, is the main source of concern in this essay.

No Stata code for the second edition of *PSM* is available on the Website for the book. A personal communication from Shenyang Guo[1] indicated that the Stata code for chapter 8 in the first edition is the reference code. A Microsoft Word document, "`Section_8.2.doc`" is available.[2] That document includes the Stata replication code for the first 4 of the 6 models that were discussed in the book. Code for Models 5 and 6 is unavailable, but that is not a material disadvantage at this point. The revised simulation estimates that will be presented for Models 1 through 4 make the point clearly enough.

## Untangling the Trouble: `corr2data`

This project is focused on replication of parameter estimates reported in Table 11.2 of *PSA* (p. 354). In Table 1, we have collected three sets of estimates using Stata 14 (using "`version 9"` emulation for backward compatability).[3] Column one in Table 1 includes the replicated estimates from *PSM.* In each cell, three numbers are reported (10,000 replications were conducted for each type):

1. The mean of the treatment effect estimates
2. The standard deviation of the estimates
3. The mean squared error (MSE), an empirical estimate of the squared difference between the estimate and the true value. As Guo and Fraser note, the MSE is equal to $bias + variance^2$, where *bias* is the expected value of the difference between the estimate and the true value.

To illustrate the meaning of the values, consider the top left cell in the table. Setting 1 is the "Selection on observables" model and Model 1 uses the ordinary least squares (OLS) estimator.

[1]November 2, 2015
[2]http://ssw.unc.edu/psa/sites/ssw.unc.edu.psa/files/Section_8.2.doc, accessed December 5, 2015
[3]Changes in Stata have altered the random generator so that estimates reported by Stata 14 differ from the published values. The discrepancy is eliminated by inserting "`version 9`" un the code.

The mean of 10,000 simulations is 0.5375, which exactly matches the estimate reported in *PSM*. The estimate of the bias of the OLS estimator is 0.0375 and the mean squared error is .0120.

The reported bias in OLS estimates is a major concern to researchers. It seems safe to say that a major source of the urgency for the implementation of new causal effect estimators is the bias of the widely used tools like OLS.

In a separate simulation that was developed in R (R Core Team, 2015), we found no apparent bias in the OLS estimates. The mismatch prompted the decision to investigate the Stata code more deeply. The Stata code for Model 1, as offered in "`Section_8.2.doc`", is included in Appendix 1. The script creates a Stata program that uses a for loop to cycle through 10,000 iterations, in each of which a (seemingly) fresh data set is manufactured and analyzed. The code for Models 2-4 is of the same design. Note on line 58, the random generator stream is re-initialized with

```
set seed 1000
```

The value 1000 is arbitrary. It is simply a user-specified integer that repositions the pseudo-random generator's position in the stream of random numbers to a given spot.[4] Randomly drawn values that depend on the Stata system-wide pseudo random number generator (PRNG) can be replicated by re-setting the seed in future runs.

The focal point of this re-examination is the usage of `corr2data` on lines 18-20.

```
corr2data x1 x2 x3 z u, /*
   */ corr(1,.2,.3,0,0\.2,1,0,0,0\.3,0,1,0,0\0,0,0,1,.4\0,0,0,.4,1) /*
   */ means (3 2 10 5 0) sds (.5 .6 9.5 2 1)
```

As we shall see, this usage of `corr2data` provides data that is neither multi-variate normal nor randomly varying. The Stata documentation for `corr2data` explains:

```
[corr2data] creates a new dataset with a specified covariance
(correlation) structure.... The purpose of this is to allow you to
perform analyses from summary statistics (correlations/covariances and
maybe the means) when these summary statistics are all you know and
summary statistics are sufficient to obtain results.

The data created by corr2data are artificial; they are not the original
data, and it is not a sample from an underlying population with the
summary statistics specified.  See [D] drawnorm if you want to generate
a random sample.  In a sample, the summary statistics will differ from
the population values and will differ from one sample to the next.

The dataset corr2data creates is suitable for one purpose only:
performing analyses when all that is known are summary statistics and
those summary statistics are sufficient for the analysis at hand
(corr2data manual, StataCorp 2015).
```

The purpose behind Stata's `corr2data` function is probably unclear to researchers in fields where individual-level data sets are the norm. On the other hand, researchers in psychology will instantly recognize the purpose. Some statistical procedures can be carried out with the sufficient statistics. For protection of individual privacy, some scholars are willing to exchange variance matrices, but not the individual level data. However, many routines in Stata are designed to receive individual level data, not covariance matrices. The `corr2data` function fills the gap, fabricating an individual

[4]It is shown below that this seed argument is ignored by the corr2data function.

level data set that is consistent with the summary statistics which can then be used with Stata procedures.

There are three features about `corr2data` that distinguish it from `drawnorm`. First, `corr2data` produces data sets that have empirical column means and observed covariance (and correlation) matrices that exactly match the user's request. The sufficient statistics do not vary between iterations. Draws from a random sampling process, as in `drawnorm`, will inevitably display variation in averages and variances. If all of the random columns in this exercise, including $v$, had been drawn from `corr2data`, then each each of the 10,000 iterations would have been identical. As it is, $v$ is drawn as a separate, uncorrelated vector within each iteration, so the estimates from each run are not exactly the same. (But the means and correlations between the other columns are identical across runs.)

Second, the data flowing from `corr2data` is not provably consistent with an $MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ random process. On the other hand, the procedure followed in `drawnorm` is entirely consistent with the recommendations of the literature about simulation of $MVN$ samples (Scheuer & Stoller, 1962; Devroye, 1986). The matrix algebra to compare these two procedures is written out in detail in a companion technical report.[5] The procedure to simulate $MVN$ data begins with a candidate matrix populated with draws from $N(0, 1)$. The candidate columns are re-scaled by multiplying by the square root of the user-specified $\boldsymbol{\Sigma}$ and adding $\boldsymbol{\mu}$. The difference in `corr2data` is that the columns are empirically rescaled in a way that might be called the multi-variate equivalent of "standardizing" the observed data[6]. While the Stata documentation is no doubt correct to say the return from `corr2data` is not a draw from $MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, it seems just as likely that it is a draw from some distribution that is in the vicinity of $MVN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

While one should not encourage students to use `corr2data` when they say they need multivariate normal data, it is difficult to believe that the results from `corr2data` are not "close enough" to $MVN$ to be serviceable in the *PSM* simulations. The data from `corr2data` looks quite a bit like an $MVN$ data set. The histograms of the marginal distributions are unimodal, the scatterplots have no unexpected patterns. Remembering that the `corr2data` is, in essence, centered and re-scaled $MVN$, it strains credulity to argue that the not-quite-$MVN$ data accounts for the anomalies. As a result, we continue our search for an explanation.

Third, running `corr2data` over and over again returns the exact same data set every time. (The values, not just the summary statistics, are identical.) The `corr2data` function has an argument, `seed()`, which is not available in `drawnorm`. If one runs `drawnorm` over and over again, unique, statistically independent draws are returned each time. In contrast, `corr2data` provides the exact same "random draw" every time unless the `seed` argument is provided. Again, according to the Stata documentation:

```
seed(#) specifies the seed of the random−number generator used to
generate data. # defaults to 0. The random numbers generated inside
corr2data do not affect the seed of the standard random−number
generator.
```

In the *PSM* simulation, all of the columns except $v$ are drawn from `corr2data`. Most of the columns in each of the 10,000 simulations are identical.[7] This feature of `corr2data` is the culprit in the mystery of the mismatch between estimates from our R implementation and the Stata

[5] "Where to Multivariate Normal Samples Come from?"

[6] Standardizing a variable: $(x_i - mean(x_i))/std.dev.(x_i)$

[7] One reader pointed out that the *PSM* program is inefficient. There is no need to recalculate with corr2data within each iteration. One can calculate that portion of the data outside the for loop, thus avoiding 10,000 additional calls to re-generate the same data.

implementation in *PSM*.

It was not necessarily so, however. The `corr2data` argument `seed` can force the output to vary. This is not an argument in favor of using `corr2data`, but rather an illustration of the point that it is not the non-normality of the return from `corr2data` that causes the estimates to appear highly biased. The problem is the similarity of the simulated data sets. We suggest using `drawnorm`, rather than `corr2data`, but if one insists on using `corr2data`, some caution about the `seed` argument is needed. There is little documentation about how one can specify `seed` so that two data sets are statistically independent from one another. If the user runs `corr2data` with `seed(994234)` and `seed(245222)`, for example, the integers seem "far apart", but we simply have no way of knowing if they may tap overlapping sections of the random number stream. Second, one notices peculiarity in the data drawn from `corr2data`. In casual testing, simulated data sets remain identical when the seed is altered.[8]

## Results

Table 1 includes three sets of estimates obtained with Stata 14 using the version 9 emulation. The replication estimates in the first column were calculated with the the original *PSM* code (with the insertion of "version 9"). Because the numerical values are identical to the published estimates in *PSM*, so we are confident we are in the right ballpark. Columns 2 and 3 are the same summaries for two variations in the Stata code. These are discussed next.

### New Estimates using `drawnorm`

The path to revise the *PSM* code is clear enough: replace `corr2data` with `drawnorm`. The new code stanza to replace the block cited previously is

```
drawnorm x1 x2 x3 z u, n(500)/*
    */ corr(1,.2,.3,0,0\.2,1,0,0,0\.3,0,1,0,0\0,0,0,1,.4\0,0,0,.4,1) /*
    */ means (3 2 10 5 0) sds (.5 .6 9.5 2 1)
```

None of the other lines in the program in Appendix I need to be revised in this exercise.[9]

The summary statistics are reported for the `drawnorm`-based simulations in the second column of Table 1. The estimates from Models 1, 2, and 4 are substantially different from the original *PSM* estimates.

[8] One will observe that identical data sets are created by using corr2data with seed(2) and seed(3) in this Stata do file:

```
set seed 1000
set obs 500
corr2data x1 x2 x3 z u , /*
*/ corr (1 ,.2 ,.3 ,0 ,0\.2 ,1 ,0 ,0 ,0\.3 ,0 ,1 ,0 ,0\0 ,0 ,0 ,1 ,.4\0 ,0 ,0 ,.4 ,1) /*
*/ means (3 2 10 5 0) sds (.5 .6 9.5 2 1) seed(2) clear
sum
corr2data x1 x2 x3 z u , /*
*/ corr (1 ,.2 ,.3 ,0 ,0\.2 ,1 ,0 ,0 ,0\.3 ,0 ,1 ,0 ,0\0 ,0 ,0 ,1 ,.4\0 ,0 ,0 ,.4 ,1) /*
*/ means (3 2 10 5 0) sds (.5 .6 9.5 2 1) seed(3) clear
sum
```

[9] The larger collection of programs in Section-8.2.doc does not run from top-to-bottom if we make this revision; the model named 131 causes a convergence error that terminates the calculations in the Stata 9 emulation. That model is not reported or discussed here, it is harmless to ignore it. In addition, to reduce the sized of the output files, the Stata command "quietly:" can be used to abbreviate the output from many commands.

Table 1: Revising *PSM* Table 11.2: Estimates of Treatment Effects

| | Estimates from 10,000 Samples* | | |
|---|---|---|---|
| Mean<br>(Standard Deviation)<br>Mean Square Error (MSE) | *PSM, 2ed, p. 354*<br>Replicated | drawnorm<br>correction | corr2data<br>seed() adjusted |
| Setting 1: Selection on Observables | | | |
| Model 1: OLS regression | 0.5375<br>(0.1028)<br>0.0120 | 0.5013<br>(0.1094)<br>0.0120 | .5000<br>(0.1088)<br>0.0129 |
| Model 2: Propensity score matching (greedy) | 0.4875<br>(0.1226)<br>0.0152 | 0.5017<br>(0.1282)<br>0.0164 | 0.5000<br>(0.1264)<br>0.0160 |
| Model 3: Treatment effect model | 1.9285<br>(0.0800)<br>2.0470 | 1.874<br>(0.1388)<br>1.9072 | 1.8800<br>(0.0908)<br>1.913 |
| Model 4: Matching estimator | 0.4531<br>(0.1464)<br>0.0237 | 0.5023<br>(0.1484)<br>0.0220 | 0.4996<br>(0.1488)<br>0.0221 |
| Setting 2: Selection on the unobservables | | | |
| Model 1: OLS regression | 0.6900<br>(0.1118)<br>0.0486 | 0.6661<br>(0.1164)<br>0.0411 | 0.6680<br>(0.1170)<br>0.0419 |
| Model 2: Propensity score matching (greedy) | 0.6464<br>(0.1345)<br>0.0395 | 0.6772<br>(0.1353)<br>0.0497 | 0.6790<br>(0.1372)<br>0.0509 |
| Model 3: Treatment effect model | 0.5036<br>(0.0221)<br>0.0005 | 0.5034<br>(0.1936)<br>0.0375 | 0.4999<br>(0.0365)<br>0.0013 |
| Model 4: Matching estimator | 0.6377<br>(0.1587)<br>0.0441 | 0.6954<br>(.1573)<br>0.1316 | 0.6978<br>(0.1589)<br>0.0643 |

*Stata 14 emulation of Stata version 9.

Stata code for *PSM* Models 5 and 6 is unavailable at the time of this writing.

This article is about simulation methodology, rather than propensity score matching. We leave it to the experts to debate the relative merits of the different statistical methods for causal treatment effects. Nevertheless, it is interesting to note the rehabilitation of ordinary least squares (OLS) regression that should flow from this re-estimation. OLS estimates of the treatment effect are unbiased and lower in variance than the greedy propensity score or matching estimators in Setting 1. Of course, that setting is intended to favor OLS, so perhaps this simply restores some order by clearing up an accidental misalignment between the simulation results and our intuition.

### New Estimates using re-seeded `corr2data`

The fact that the data returned from `corr2data` is not *MVN* is an eye-catcher in the Stata documentation. However, our intuition was that the data from `corr2data` ought to be *close enough* to *MVN*. The distributional difference should not be sufficient to account for the OLS bias.

We put that intuition to the test by revising the code to insert new values of the `seed` argument in `corr2data`. The revised code stanza is as follows:

```
local myseed = 50*`i' + 200
corr2data x1 x2 x3 z u, /*
   */ corr(1,.2,.3,0,0\.2,1,0,0,0\.3,0,1,0,0\0,0,0,1,.4\0,0,0,.4,1) /*
   */ means (3 2 10 5 0) sds (.5 .6 9.5 2 1) seed(`myseed')
```

The estimates are reported in the third column of Table 1. Forcing `corr2data` to return a fresh block of data with each iteration results in a shift of the estimates that is comparable to the difference in results from `drawnorm`. Readers will notice that the apparent bias of OLS has disappeared, just as in column 2. The estimates from Models 1 and 2 appear to be equally good (perhaps OLS is slightly preferable), while the matching estimator in Model 4 has similar bias but much higher variance. As in the analysis of *PSM,* in Setting 1, the treatment effect model, Model 3, generates estimates that are wildly incorrect. The only difference we note in columns two and three of Table 1 is that the variance of the Model 3 estimates is concealed by the usage of `corr2data`. In Setting 2, the estimates indicate that models 1, 2, and 3 are all biased on the high side, whereas model 3 is clearly more likely to be in the correct range. However, it appears worth mentioning that the drawnorm-based estimates in column two show much higher variance for Model 3, so that its mean squared error is not nearly so tiny compared to the other methods.

## Summary

The Monte Carlo simulations reported in *Propensity Score Matching* have been replicated. The estimates originally published have been verified. Two variations on the data generation model have been implemented.

The main conclusion should be that the use of Stata's `corr2draw` function had some deleterious impacts on the original simulation. It seems likely that the original *PSM* report overstated estimation bias in several models. Because the simulation data did not vary between iterations, the distortions caused in one random sample were propogated to all of the succeeding runs. The original *PSM* simulation amounts to an extremely rigorous re-examination of the impact of random variation in one column while the rest of the columns are exactly the same in all iterations.

For broader take-away points, we would offer the following ideas. First, it is extremely valuable to have open code for replication of simulation results. We were unable to verify the *PSM* results for the OLS model with a program written in R. It would have been very difficult or impossible to unravel the mystery without access to the Stata source code. The description of a simulation in a publication cannot generally include detail required for a full replication. The code itself is necessary.

Second, before placing a great deal of weight on an particular Monte Carlo simulation, it might be wise to obtain a similar set of results with a program written from scratch in a different language or programming environment. While it is possible for 2 programmers to make mistakes, it seems unlikely that they will make the same mistakes. The mis-match between R and Stata results inspired this investigation, but it also suggests we should continue along these lines. Since the models described in *PSM* are creatures of software and details congealed in Stata code, parallel implementations in other languages will help us feel more confident that the benefits of propensity score corrections are real and meaningful.

## Appendix 1. Guo and Fraser Simulation Code for Model 1

```
//Guo and Fraser Stata Syntax for Section 8.2, Chapter 8

//Models 1.1 and 1.1.1 (Selection on the observables, OLS with and without z)
//______________________________________________________________________
//Section 8.2 Models 1.1 and 1.1.1 OLS
//u & z correlated, u & v NOT correlated
//cd "D:\Sage\Web\Section 8.2"
capture log close
set more off
log using Model_11_&111, replace

clear
program model1_1
   postfile sim rzu ruv rwu rzv b1 t1 b2 t2 b3 t3 using sim1_1, replace
   forvalues i = 1/10000 {
     drop _all
     set obs 500
     corr2data x1 x2 x3 z u, /*
       */ corr(1,.2,.3,0,0\.2,1,0,0,0\.3,0,1,0,0\0,0,0,1,.4\0,0,0,.4,1)/*
       */ means (3 2 10 5 0) sds (.5 .6 9.5 2 1)
     gen v=invnormal(uniform())
     gen w=.5*z+.1*x3+v
     quietly: sum w,detail
     gen ww=1
     replace ww=0 if w<= r(p50)
     gen y=100+.5*x1+.2*x2-.05*x3+.5*ww+u
     quietly: correlate z u,means
         gen rzu=r(rho)
     quietly: correlate u v,means
         gen ruv=r(rho)
     quietly: correlate ww u,means
         gen rwu=r(rho)
     quietly: correlate z v,means
         gen rzv=r(rho)
     quietly: regress y x1 x2 x3 z ww
         mat v=vecdiag(e(V))
         mat b1=el(e(b),1,5)
         mat t1=el(e(b),1,5)/sqrt(el(v,1,5))
          svmat b1
          svmat t1
      quietly:regress y x1 x2 x3 ww
         mat v=vecdiag(e(V))
```

```
          mat b2=el(e(b),1,4)
          mat t2=el(e(b),1,4)/sqrt(el(v,1,4))
           svmat b2
           svmat t2
       quietly: regress y x1 x2 ww
          mat v=vecdiag(e(V))
          mat b3=el(e(b),1,3)
          mat t3=el(e(b),1,3)/sqrt(el(v,1,3))
           svmat b3
           svmat t3
          post sim (rzu) (ruv) (rwu) (rzv) (b1) (t1) (b2) (t2) (b3) (t3)
        }
      postclose sim
end

set seed 1000
model1_1

use sim1_1, replace
sum rzu ruv rwu b1 b2

quietly: sum b1
display r(sd)^2
quietly: sum b2
display r(sd)^2

gen v1= (b1-.5)^2
quietly:  sum v1
gen mse1=r(sum)/10000
gen v2= (b2-.5)^2
quietly: sum v2
gen mse2=r(sum)/10000

sum mse1 mse2

log close
```

## References

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer, 1986 edition edition.

Guo, S. & Fraser, M. W. (2009). *Propensity Score Analysis: Statistical Methods and Applications*. Los Angeles: SAGE Publications, Inc, 1ed edition.

Guo, S. & Fraser, M. W. (2015). *Propensity Score Analysis: Statistical Methods and Applications*. Los Angeles: SAGE Publications, Inc, 2ed edition.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Scheuer, E. M. & Stoller, D. S. (1962). On the Generation of Normal Random Vectors. *Technometrics*, 4(2), 278.

StataCorp (2015). *Stata 14 Base Reference Manual*. Stata Press., College Station, TX.