Stata syntax for the Monte Carlo simulation used in Zhang, Z., Zyphur, M. J. & Preacher, K. J. (In press). Testing multilevel mediation using hierarchical linear models: Problems and solutions. *Organizational Research Methods.*

The following is one of the simulation .do files using Stata v.9.0. It is not for analyzing a real data set for hypothesis testing. If you need to analyze a real data set, please implement the CWC(M) equations in Table 1 of the paper using your own software, e.g., the HLM software, SAS Proc Mixed, or Stata.

```
/* Monte Carlo simulation */
/* Zhang, Zyphur, & Preacher (in press) Org. Research Methods */
/* Generate 500 two-level data sets, Total N=600 */
/* Then, analyze mediation effects */
/* Last update Oct. 9, 2008 */

/* Key parameters need to be changed for each cell of the simulation */
/*** size 12 * 50 groups      ***/
/*** between coefficient= .14 ***/
/*** within coefficient= -.39 ***/

#delimit;
set more off;
version 9;
clear;
capture log close;

local i=1;
while `i'<= 500 {;
/* Level 2 data generation */
matrix Cor2 = (1, .40 \ .40, 1);
drawnorm x m_bar, n(50) corr(Cor2) ;   /* # of groups=50 */
gen u0j = .2236068 * invnormal(uniform());  /*level 2 error var=.05 */
gen id=_n;
sort id;
save rep_`i'_bar.dta, replace;
clear;
/* Level 1 data generation */
set obs 600;
gen m_1 = invnormal(uniform());
gen rij= .5* invnormal(uniform());   /* level 1 error term var =.25 */
gen id =int((_n-1)/12)+1 ;           /* group size = 12 */
sort id;
merge id using rep_`i'_bar.dta;
gen m=m_1 + m_bar;
/*model specification*/
gen y= 0 + .3*x + .14*m_bar + u0j + (-.39)*m_1 + rij;
drop  _merge;
save rep_`i'.dta, replace; /* save as separate data sets */
clear;
local i= `i' + 1;
};

/* Start to analyze data */
local p =1;
postfile myfile  id g01_1 g01_1v g01_2 g01_2v b1j_3 b1j_3v g01_3 g01_3v
```

```
b1j_4 b1j_4v g01_4 g01_4v g02_4 g02_4v corr_x_m using finaldata.dta, replace;
quietly while `p'<= 500 {;  /*“quietly” suppress screen output in this step*/
use rep_`p'.dta, clear;
/* Step 1 in Baron & Kenny (1986) */
xtmixed y x || id: ;
matrix b1=e(b);
matrix v1=e(V);
matrix g01_1 =b1[1..1, 1..1];
matrix g01_1v =v1[1..1, 1..1];

/* Step 2 B&K */
xtmixed m x || id:;
matrix b2=e(b);
matrix v2=e(V);
matrix g01_2 =b2[1..1, 1..1];
matrix g01_2v =v2[1..1, 1..1];

/* Step 3 Grand-mean centering or raw metric */
xtmixed y  m x || id:;
matrix b3=e(b);
matrix v3=e(V);
matrix b1j_3 =b3[1..1, 1..1];
matrix b1j_3v = v3[1..1, 1..1];
matrix g01_3 = b3[1..1, 2..2];
matrix g01_3v = v3[2..2, 2..2];

/* Step 4, group-mean centering */
xtmixed y  m_1 x m_bar || id:;
matrix b4=e(b);
matrix v4=e(V);
matrix b1j_4=b4[1..1, 1..1];
matrix b1j_4v= v4[1..1, 1..1];
matrix g01_4= b4[1..1, 2..2];
matrix g01_4v = v4[2..2, 2..2];
matrix g02_4=b4[1..1, 3..3 ];
matrix g02_4v=v4[3..3, 3..3];

/* Total raw correlation between x and m */
corr x m;

post myfile (`p') (trace(g01_1)) (trace(g01_1v)) (trace(g01_2))
(trace(g01_2v)) (trace(b1j_3)) (trace(b1j_3v)) (trace(g01_3)) (trace(g01_3v))
(trace(b1j_4)) (trace(b1j_4v)) (trace(g01_4)) (trace(g01_4v)) (trace(g02_4))
(trace(g02_4v)) (r(rho));
local p= `p' + 1;
};
postclose myfile;
clear;

use finaldata.dta, clear;
/* Analyzing biased abw_product, Sobel 1982 first-order solution */
/* abw refers to the “wrong” estimate of ab */
gen abw_prod = g01_2 * b1j_3;
gen abw_prod_z = (g01_2 * b1j_3)/(((g01_2)^2* b1j_3v+(b1j_3)^2*g01_2v)^(.5));
gen abw_prod_z_abs = abs(abw_prod_z );
gen abw_prod_dummy = abw_prod_z_abs > 1.96;
egen abw_prod_sum = total(abw_prod_dummy) ; /*type I error rate or power */
gen abw_type1 = abw_prod_sum /500;
```

```
/* Analyzing true abt_product */
/* abt refers to the “true” estimate of ab */
gen abt_prod = g01_2 * g02_4;
gen abt_prod_z = abt_prod /(((g01_2)^2*g02_4v+ (g02_4)^2*g01_2v )^(.5));
gen abt_prod_z_abs = abs(abt_prod_z );
gen abt_prod_dummy = abt_prod_z_abs > 1.96;
egen abt_prod_sum = total(abt_prod_dummy);  /* Type I error rate or power*/
gen abt_type1 = abt_prod_sum / 500;

egen point_t = mean(abt_prod);
egen point_w = mean(abw_prod);
gen pt_bias = (point_w - point_t)*100/point_t;
egen sobel_t = mean(abt_prod_z );
egen sobel_w = mean(abw_prod_z );
gen sol_bias = (sobel_w - sobel_t)*100/ sobel_t;

log using b14w_39.log, replace;     /*Save into a log file */

list id g01_2 g02_4 g01_3 b1j_3 b1j_4 in 1/5, table;
list point_t point_w pt_bias sobel_t sobel_w sol_bias abt_type1 abw_type1  in
1/4, table;

/* Analyzing Freedman and Schatzkin, 1992, need raw score corr X and M */
/* Analyzing biased c-c' Freedman and Schatzkin, 1992 */
gen ccw_minus = g01_1 - g01_3;
gen ccw_minus_z = (g01_1 - g01_3)/ ((g01_1v + g01_3v -2*((g01_1v* g01_3v*(1-
(corr_x_m)^2) )^(.5)))^(.5));
gen ccw_minus_z_abs = abs(ccw_minus_z);
gen ccw_minus_dummy = ccw_minus_z_abs >1.96;
egen ccw_minus_sum = total (ccw_minus_dummy );
gen ccw_type1 = ccw_minus_sum / 500;

/* unbiased c-c' */
gen cct_minus = g01_1 - g01_4;
gen cct_minus_z = (g01_1 - g01_4)/(( g01_1v + g01_4v -2* (( g01_1v*
g01_4v*(1- (corr_x_m)^2) )^(.5)) )^(.5));
gen cct_minus_z_abs = abs(cct_minus_z);
gen cct_minus_dummy = cct_minus_z_abs >1.96;
egen cct_minus_sum = total (cct_minus_dummy );
gen cct_type1 = cct_minus_sum / 500;

egen cc_t = mean(cct_minus);
egen cc_w = mean(ccw_minus);
gen cc_bias = (cc_w - cc_t)*100/cc_t;
egen ccz_t = mean(cct_minus_z );
egen ccz_w = mean(ccw_minus_z );
gen ccz_bias = (ccz_w - ccz_t)*100/ ccz_t;

list cc_t cc_w cc_bias ccz_t ccz_w ccz_bias cct_type1 ccw_type1 in 1/5, table;
gen disc_group = ( cc_t - point_t ) * 100/ point_t ;
gen disc_grand =( cc_w - point_w) *100 / point_w ;
list disc_group disc_grand in 1/5 ;
log close;
save b14w_39.dta, replace;
exit;
```