



JENA ECONOMIC RESEARCH PAPERS



2013 – 027

Endogenous variables in non-linear models with mixed effects: Inconsistence under perfect identification conditions?

by

**Franz Buscha
Anna Conte**

www.jenecon.de

ISSN 1864-7057

The JENA ECONOMIC RESEARCH PAPERS is a joint publication of the Friedrich Schiller University and the Max Planck Institute of Economics, Jena, Germany. For editorial correspondence please contact markus.pasche@uni-jena.de.

Impressum:

Friedrich Schiller University Jena
Carl-Zeiss-Str. 3
D-07743 Jena
www.uni-jena.de

Max Planck Institute of Economics
Kahlaische Str. 10
D-07745 Jena
www.econ.mpg.de

© by the author.

Endogenous variables in non-linear models with mixed effects: Inconsistence under perfect identification conditions?

Franz Buscha^{1,*} and Anna Conte^{1,2}

¹ *Westminster Business School, University of Westminster, 35 Marylebone Road, London, NW1 5LS, United Kingdom*

² *Strategic Interaction Group, Max Planck Institute of Economics, Kablaische strasse 10, 07745 Jena, Germany*

Abstract

This paper examines the consequences of introducing a normally distributed effect into a system where the dependent variable is ordered and the explanatory variable is ordered and endogenous. Using simulation techniques we show that a naïve bivariate ordered probit estimator which fails to take a mixed effect into account will result in inconsistent estimates even when identification conditions are optimal. Our results suggest this finding only applies to non-linear endogenous systems.

Keywords: bivariate probit, bivariate ordered probit, mixed effects, endogenous binary variables, constant parameters.

JEL classification: C35, C36, C51

* Tel: +44 (0)20 7911 5000 x66596
E-Mail address: buschaf@wmin.ac.uk

1. Introduction

When an endogenous explanatory variable (or suspected as such) is encountered in empirical economic applications, instrumental variable (IV) methods are routinely used to estimate a causal and consistent effect. However, when both the dependent variable and the endogenous variable take the form of binary or categorical data, standard IV techniques (such as two-stage least squares) often fail and more sophisticated analytical techniques are needed. In such cases, bivariate probit models can be used when both dependent and endogenous explanatory variables are binary, whilst bivariate ordered probit models can be used when either (or both) variables take an ordered form (Greene and Hensher, 2010: 290-296).

Many applications of linear or non-linear IV models generally assume that the impact of the endogenous variable is constant over all individuals. This, however, may not necessarily be the case as recent work has shown (Buscha and Conte, 2013). The introduction of a mixed effect - approach emerging in the behavioural economics literature - queries the assumption of constant parameters over all individuals and argues that in many cases behavioural relationships are better represented by distributional or heterogeneous effects, measured by β_i - the parameter specific to individual i - as opposed to β , which is assumed constant over the population. The idea we want to convey here is that some relationships may be partly influenced by unobservable factors (such as ability, motivation, satisfaction, beliefs, etc.) Our mixed effects approach in the context of bivariate models is meant to capture the heterogeneous effect of such unobservable factors.

In this paper, we highlight the repercussions that the constant parameter assumption can have in a linear and non-linear framework when the effect of the endogenous variable on the dependent variable is not constant across the population but follows a certain distribution. Specifically, we show that, when the effect of an endogenous variable is normally distributed, linear IV techniques that do not take into account this distributional effect will at least recover a consistent average effect. The same cannot be said for non-linear models which exhibit inconsistency when the effect is heterogeneous but treated as constant. This finding has potential repercussions for empirical analysis.

2. Endogenous models with a distributional effect

Consider the following linear system:

$$\begin{cases} y_{1i} = 1x_{1i} + 1z_{1i} + \varepsilon_{1i} \\ y_{2i} = \gamma_i y_{1i} + 1x_{1i} + \varepsilon_{2i} \end{cases} \quad (1)$$

Here, y_{1i} appears as a regressor in the equation for y_{2i} and the causal effect of y_{1i} on y_{2i} is measured by $\gamma_i \sim N(\mu_\gamma = 1, \sigma_\gamma = 1)$. That is, across the population there exists an average effect of y_{1i} on y_{2i} of 1 with a unit standard deviation. x_{1i} and z_{1i} are explanatory variables drawn from two independent standard normals, and ε_{1i} and ε_{2i} are jointly normally distributed error terms with correlation ρ . Let us assume, for the time being, that $\rho = 0.5$. Given these hypotheses, we can think of y_{1i} as an endogenous regressor, since it is correlated with the error term ε_{2i} .

The above system is generally handled via IV techniques, with z_{1i} acting as an exclusion restriction and thus formally identifying the system. Under the condition that z_{1i} is not correlated with the error term ε_{2i} , a causal effect of y_{1i} on y_{2i} can be consistently estimated.

Now, consider the same system under non-linear specifications of the dependent variables:

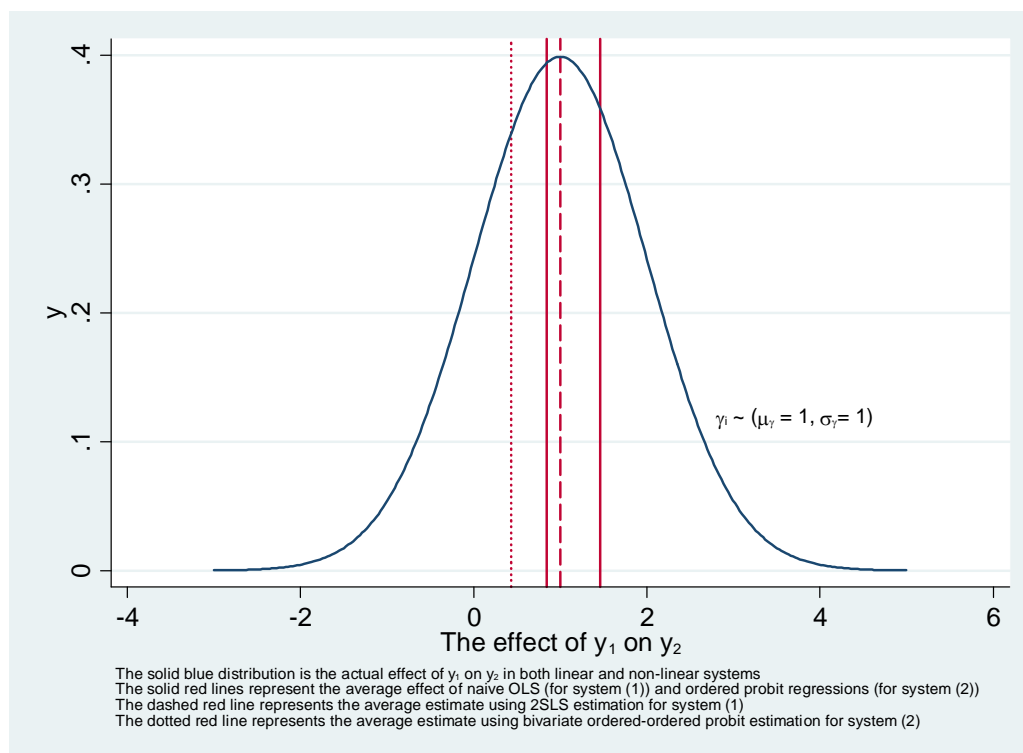
$$\begin{cases} y_{1i}^* = 1x_{1i} + 1z_{1i} + \varepsilon_{1i} \\ y_{2i}^* = \gamma_i y_{1i}^* + 1x_{1i} + \varepsilon_{2i} \end{cases} \quad (2)$$

Here, the latent variables y_{1i}^* and y_{2i}^* are subject to the following observational rule:

$$y_{1i} = \begin{cases} 1 & \text{if } y_{1i}^* \leq -1 \\ 2 & \text{if } -1 < y_{1i}^* \leq 0 \\ 3 & \text{if } 0 < y_{1i}^* \leq 1 \\ 4 & \text{if } 1 < y_{1i}^* \end{cases} \quad y_{2i} = \begin{cases} 1 & \text{if } y_{2i}^* \leq -1 \\ 2 & \text{if } -1 < y_{2i}^* \leq 0 \\ 3 & \text{if } 0 < y_{2i}^* \leq 1 \\ 4 & \text{if } 1 < y_{2i}^* \end{cases} \quad (3)$$

This system requires a bivariate ordered probit type methodology to successfully recover the effect of y_{1i} on y_{2i} (see for example, Sajaia, 2008). However, in both (1) and (2) the effect of y_{1i} on y_{2i} , γ_i , is non-constant. We will first simulate the effect of estimating both (1) and (2) without taking into account the heterogeneous effect that y_{1i} has on y_{2i} . We run 1000 replications with simulated samples of 1000 observations each for both (1) and (2), and graph these results in Figure 1 to visualise the effect of such naïve estimates.

Figure 1: Non-mixed estimation of a mixed effect



As expected, simulation results in Figure 1 show that, when the endogenous nature of y_{1i} is not taken into account, inconsistency may result. OLS and ordered probit models cannot recover a true causal effect in an endogenous system. However, even when the endogenous nature of y_{1i} is taken into account, the absence of a mixed effect in naïve estimation cannot reveal the true average impact that y_{1i} has on y_{2i} . Hence, at least an average effect of the distributional effect can be consistently estimated. Moreover, results from our simulations suggest that in the non-linear system, even under perfect identification conditions, the estimated average effect exhibits inconsistency. Only the linear IV framework can successfully recover an average effect of 1.

3. Monte Carlo Simulations

To investigate this phenomenon further, we devise a series of Monte Carlo simulations where we change parameter values in the above systems so that $\rho \in \{-0.9, -0.5, 0, 0.5, 0.9\}$. The effect of y_{1i} on y_{2i} is $\gamma_i \sim N(\mu_\gamma = 1, \sigma_\gamma = 1)$. We run 1,000 replications for sample size 1,000 and estimate four separate models: 2SLS regression for system (1); maximum likelihood IV regression with a mixed effect for system (1); naïve bivariate ordered-ordered probit regression for system (2); and bivariate ordered-ordered probit regression with a mixed effect for system (2).

Table 1: Monte Carlo Simulations of linear and non-linear endogenous systems

Simulations when $\gamma_i \sim N(\mu_\gamma = 1, \sigma_\gamma = 1)$							
System (1)	<i>2SLS without distributional effect</i>			<i>IV with distributional effect</i>			
Rho	μ_γ	σ_γ	ρ	Rho	μ_γ	σ_γ	ρ
-0.9	0.999	n/a	n/a	-0.9	1.003	0.975	n/a
-0.5	1.000	n/a	n/a	-0.5	1.002	0.986	n/a
0	1.000	n/a	n/a	0	1.001	0.994	n/a
0.5	1.000	n/a	n/a	0.5	1.000	0.991	n/a
0.9	0.999	n/a	n/a	0.9	0.999	0.993	n/a
System (2)	<i>Bivariate ordered probit without distributional effect</i>			<i>Bivariate ordered probit with distributional effect</i>			
Rho	μ_γ	σ_γ	ρ	Rho	μ_γ	σ_γ	ρ
-0.9	0.440	n/a	-0.529	-0.9	0.993	0.988	-0.901
-0.5	0.448	n/a	-0.238	-0.5	1.072	1.032	-0.539
0	0.444	n/a	0.101	0	1.060	1.052	-0.065
0.5	0.431	n/a	0.442	0.5	1.043	1.095	0.561
0.9	0.411	n/a	0.725	0.9	0.975	0.981	0.929

Monte Carlo Simulations: 1000 Replications. Values are averages.

Table 1 presents the average estimates of μ_γ , σ_γ and ρ (when estimable). We see that in the linear framework (1), 2SLS estimates the correct value of μ_γ . Due to its simplicity, no other parameters are recovered. A more complex two-stages OLS - maximum likelihood IV estimator, which takes into account the distribution around μ_γ , recovers the value taken by the σ_γ parameter.

However, in the non-linear case (2), dealing with endogeneity via a bivariate ordered probit model can result in significantly different estimates depending on whether the heterogeneity of γ_i is taken into account. In this regard, Table 1 shows that, when the heterogeneity of γ_i is not taken into account, estimates of μ_γ and ρ exhibit substantial inconsistency. Such inconsistency is more severe for μ_γ than for ρ . Only when the heterogeneity of γ_i is considered, the bivariate ordered probit model *does* recover consistent estimates of μ_γ , σ_γ and ρ .

The above results are non-trivial. The assumption of constant parameters over all individuals in behavioural or micro-economic research is unlikely to hold in many instances. For example, the assumption of a normally distributed effect - where an effect is symmetric around a mean and the majority of individuals are ‘clumped’ near this mean whilst a few individuals experience extreme positive or negative effects - is probably a reasonable representation of the effect of many complicated behavioural decisions. Our example clearly demonstrates that, when such decisions are realised as categorical choices, endogenous non-linear models which fail to account for a mixed effect can lead to substantially inconsistent estimates, even under perfect identification conditions.

Table 2: Monte Carlo Simulations of a non-linear endogenous system under imperfect identification

Simulations when $\gamma_i \sim N(\mu_\gamma = 1, \sigma_\gamma = 1)$					
System (2)	<i>Bivariate ordered probit without mixed effect</i>				
	Estimates of μ_γ				
Rho	$\theta = 0$	$\theta = 0.01$	$\theta = 0.1$	$\theta = 0.5$	$\theta = 1$
-0.9	0.440	0.445	0.493	0.666	0.709
-0.5	0.448	0.454	0.507	0.705	0.795
0	0.444	0.450	0.508	0.708	0.777
0.5	0.431	0.439	0.502	0.674	0.676
0.9	0.411	0.419	0.486	0.621	0.539
	<i>Bivariate ordered probit with mixed effect</i>				
	Estimates of μ_γ				
Rho	$\theta = 0$	$\theta = 0.01$	$\theta = 0.1$	$\theta = 0.5$	$\theta = 1$
-0.9	0.993	0.988	0.987	1.071	1.193
-0.5	1.072	1.202	1.172	1.259	1.580
0	1.060	1.093	1.259	1.530	1.596
0.5	1.043	1.108	1.247	1.629	1.777
0.9	0.975	1.028	1.228	1.818	2.272
	Estimates of σ_γ				
Rho	$\theta = 0$	$\theta = 0.01$	$\theta = 0.1$	$\theta = 0.5$	$\theta = 1$
-0.9	0.988	0.902	0.871	0.611	0.662
-0.5	1.032	1.095	1.189	0.962	1.046
0	1.052	1.103	1.217	1.203	1.079
0.5	1.095	1.119	1.178	1.267	1.262
0.9	0.981	1.022	1.131	1.288	1.589

Monte Carlo Simulations: 1000 Replications. Values are averages.

However, perfect identification relies on the theoretical assumption that the instrument in question is not correlated with the error term in the explanatory equation (ε_{2i}). This assumption cannot be proven and in reality may often fail, i.e., a partial correlation between the instrument z_{1i} and the error term ε_{2i} may exist. Under such conditions, estimates of the mean of γ_i, μ_γ , are inconsistent; however, it is uncertain to what extent estimates of σ_γ may be affected. To investigate this issue we devise a further set of Monte Carlo simulations for system (2) where we define a correlation between the instrument z_{1i} and the error term ε_{2i} of $\text{corr}(z_{1i}, \varepsilon_{2i}) = \theta$ and $\theta \in \{0, 0.01, 0.1, 0.5, 1\}$. When $\theta = 0$, identification conditions are perfect whilst higher values of θ will lead to larger inconsistencies.

Simulations from Table 2 show that as the instrument becomes ‘weaker’, inconsistencies in both type of estimators appears. Stronger correlations between the instrument z_{1i} and the error term ε_{2i} result in larger inconsistencies. As before, the naïve bivariate ordered probit estimator produces inconsistent estimates under all identification conditions, whilst the bivariate ordered probit estimator with a mixed effect only produces inconsistent estimates when identification is imperfect. Evidence also suggests that both μ_γ and σ_γ exhibit inconsistency under imperfect identification conditions, that is to say that there are no “freebies” in this type of analysis. Perfect identification conditions with the appropriate distributional modelling are a prerequisite for causal inference when challenged with non-linear endogenous systems.

4. Conclusion

A recent paper by Chiburis *et al.* (2012) argues that the existing literature has offered relatively little advice when deciding on whether binary endogenous structures ought to be estimated by linear IV or bivariate probit methods. Using simulation techniques, they aim to offer practical advice in helping researchers make an informed decision on what type of model to choose. In a similar vein, the results presented here expand this choice set to encompass heterogeneous effects and ought to be seen as practical advice to empirical researchers. Our results suggest that there is a distinct possibility that many studies which make use of bivariate (ordered) probit type methodologies may recover incorrect causal parameters, if the underlying process includes a heterogeneous effect. We urge applied researchers to take this possibility into consideration.

It should be noted that our results have not been generalised towards any distributional form. Our results are only applicable when the underlying mixed effect follows a normal distribution, although we suspect our results are applicable to other distributions (such as log-normal or power).

Appendix

Simulation results for 2SLS were carried out using the **ivregress** command in Stata. Simulation results for the bivariate ordered probit without mixed effects were carried out using the **bioprobit** command in Stata written by Sajaia (2008). Simulations for the linear IV estimator with mixed effects and the bivariate ordered probit estimator with mixed effects were written as Stata .do files and are available at from the authors or in the appendix of this working paper.

References

Buscha, F. and Conte, A. (2013) “The Impact of Truancy on Educational Attainment during Compulsory Schooling: a Bivariate Ordered Probit Estimator with Mixed Effects”, *The Manchester School*, available online, early view.

Chiburis, R. and Das, J. and Lokshin, M. (2012) “A practical comparison of the bivariate probit and linear IV estimators”, *Economics Letters*, Vol. 117, pp. 762-766

Greene, W.H. and Hensher, D.A. (2010) *Modelling Ordered Choices: A Primer*, Cambridge University Press, New York

Sajaia, Z. (2008) “Maximum Likelihood Estimation of a Bivariate Ordered Probit Model: Implementation and Monte Carlo Simulations”, World Bank Applied Economic Research Papers

Stata Code

Stata code to estimate system (1)

```

*****
cap log close
log using ivmixed.log, replace

program drop _all

program sm_ml_lf
    args lnf xb2 mg sg lnsd2

    tempvar l X1 X2 rho sigmag g L1 L2 sd1 sd2 eta rt s1 s2

    quietly{
        /*components to be indirectly estimated*/
        gen double `sigmag'= exp(`sg')
        gen double `sd2'= exp(`lnsd2')

        gen double `X1'=0
        gen double `X2'=0
        gen double `g'=0
        gen double `eta'=0
        gen double `rt'=0
        gen double `s1'=0
        gen double `s2'=0

        /*partial likelihoods*/
        gen double `l'=0
        gen double `L1'=0
        gen double `L2'=0

        forvalues r=1/`$draws'{
            quietly {
                replace `g'=`mg'+sqrt(2)*`sigmag'*a`r'

                replace `X2'=`g'*y1star+`xb2'

                replace `l'=`normalden($ML_y1,`X2',`sd2')

                replace `L1'=(1/sqrt(2*asin(1)))*w`r'*`l'
                replace `L2'=`L2'+`L1'
            }
        }

        quietly replace `lnf'=ln(`L2')
    }

exit

end

program define myprog2, rclass
    syntax [, obs(integer 1) rho(real 0) mg(real 0) sg(real 0)]

clear

```

```

set obs `obs'
set more off

scalar rho=`rho'
scalar sd1=.5
scalar sd2=.5
scalar mg=`mg'
scalar sg=`sg'

    *generate covariates*
    gen x1 = rnormal()
    gen z1 = rnormal()
    gen u1=rnormal()
    gen u2=rnormal()
    gen e1=sd1*u1
    gen e2=sd2*(rho*u1+sqrt(1-rho^2)*u2)

    gen gamma = mg+sg*rnormal()

    gen y1 =          + 1*x1 + 1*z1 + e1
    gen y2 = gamma*y1 + 1*x1 + e2

    reg y1 x1 z1, noconst
    predict y1star, xb

gen double a1=.
replace a1=-2.453407083009012499e-01
gen double a2=.
replace a2=-1.234076215395323007e+00
gen double a3=.
replace a3=-2.254974002089275523e+00
gen double a4=.
replace a4=-3.347854567383216326e+00
gen double a5=.
replace a5=-4.603682449550744272e+00
gen double a6=.
replace a6=-7.374737285453943587e-01
gen double a7=.
replace a7=-1.738537712116586206e+00
gen double a8=.
replace a8=-2.788806058428130480e+00
gen double a9=.
replace a9=-3.944764040115625210e+00
gen double a10=.
replace a10=-5.387480890011232861e+00
gen double a11=.
replace a11=2.453407083009012499e-01
gen double a12=.
replace a12=1.234076215395323007e+00
gen double a13=.
replace a13=2.254974002089275523e+00
gen double a14=.
replace a14=3.347854567383216326e+00
gen double a15=.
replace a15=4.603682449550744272e+00
gen double a16=.
replace a16=7.374737285453943587e-01
gen double a17=.
replace a17=1.738537712116586206e+00
gen double a18=.
replace a18=2.788806058428130480e+00
gen double a19=.

```

Jena Economic Research Papers 2013 - 027

```
replace a19=3.944764040115625210e+00
gen double a20=.
replace a20=5.387480890011232861e+00

gen double w1=.
replace w1=4.622436696006100896e-01
gen double w2=.
replace w2=1.090172060200233200e-01
gen double w3=.
replace w3=3.243773342237861832e-03
gen double w4=.
replace w4=7.802556478532063693e-06
gen double w5=.
replace w5=4.399340992273180553e-10
gen double w6=.
replace w6=2.866755053628341297e-01
gen double w7=.
replace w7=2.481052088746361088e-02
gen double w8=.
replace w8=2.283386360163539672e-04
gen double w9=.
replace w9=1.086069370769281693e-07
gen double w10=.
replace w10=2.229393645534151292e-13
gen double w11=.
replace w11=4.622436696006100896e-01
gen double w12=.
replace w12=1.090172060200233200e-01
gen double w13=.
replace w13=3.243773342237861832e-03
gen double w14=.
replace w14=7.802556478532063693e-06
gen double w15=.
replace w15=4.399340992273180553e-10
gen double w16=.
replace w16=2.866755053628341297e-01
gen double w17=.
replace w17=2.481052088746361088e-02
gen double w18=.
replace w18=2.283386360163539672e-04
gen double w19=.
replace w19=1.086069370769281693e-07
gen double w20=.
replace w20=2.229393645534151292e-13

global draws "20"

mat start=( 1, .3, .2, -.36)
ml model lf sm_ml_lf (y2 = x1 , noconstant) (mg:) (sg:) (lnsd2:), technique(bhhh)
ml init start, copy
ml max, iterate(200) difficult trace

nlcom (sigmag: exp([sg]_b[_cons])) ///
      (sd2: exp([lnsd2]_b[_cons]))

gen mg = ([mg]_b[_cons])
gen sg = exp([sg]_b[_cons])

end

simulate mg sg , reps(1000) : myprog2, obs(1000) rho(-0.9) mg(1) sg(1)
```

```

su
simulate mg sg , reps(1000) : myprog2, obs(1000) rho(-0.5) mg(1) sg(1)
su
simulate mg sg , reps(1000) : myprog2, obs(1000) rho(0) mg(1) sg(1)
su
simulate mg sg , reps(1000) : myprog2, obs(1000) rho(0.5) mg(1) sg(1)
su
simulate mg sg , reps(1000) : myprog2, obs(1000) rho(0.9) mg(1) sg(1)
su
*****

```

Stata code to estimate system (2)

```

*****
cap log close
log using biopmixed.log2, replace

program drop _all

program sm_ml_lf
    args lnf xb1 c11 c12 c13 c14 xb2 c21 c22 c23 c24 mg sg r

    tempvar ll Y1 Y2 lj rho eta rt sigmag g L1 L2

    quietly{
        /*components to be indirectly estimated*/
        gen double `rho'= tanh(`r')
        gen double `sigmag'= exp(`sg')

        gen double `Y1'=0
        gen double `Y2'=0
        gen double `eta'=0
        gen double `rt'=0
        gen double `g'=0

        /*partial likelihoods*/
        gen double `ll'=0
        gen double `L1'=0
        gen double `L2'=0

        forvalues r=1/$draws{
            quietly {
                replace `g'=`mg'+sqrt(2)*`sigmag'*a`r'
                replace `eta'=1/sqrt(1+2*`g'*`rho'+`g'^2)
                replace `rt'=`eta'*(`g'+`rho')

                replace `Y1'=`xb1'
                replace `Y2'=`g'*`Y1'+`xb2'

                replace `ll' = (binormal((`c11'-`Y1'),`eta'*(`c21'-`Y2'),`rt')) if m0==1
                replace `ll' = (binormal((`c11'-`Y1'),`eta'*(`c22'-`Y2'),`rt')-binormal((`c11'-`Y1'),`eta'*(`c21'-`Y2'),`rt')) if m1==1
                replace `ll' = (binormal((`c11'-`Y1'),`eta'*(`c23'-`Y2'),`rt')-binormal((`c11'-`Y1'),`eta'*(`c22'-`Y2'),`rt')) if m2==1
                replace `ll' = (binormal((`c11'-`Y1'),`eta'*(`c24'-`Y2'),`rt')-binormal((`c11'-`Y1'),`eta'*(`c23'-`Y2'),`rt')) if m3==1
                replace `ll' = ((normal((`c11'-`Y1'))-binormal((`c11'-`Y1'),`eta'*(`c24'-`Y2'),`rt')) if m4==1

                replace `ll' = (binormal((`c12'-`Y1'),`eta'*(`c21'-`Y2'),`rt')-binormal((`c11'-`Y1'),`eta'*(`c21'-`Y2'),`rt')) if m5==1
            }
        }
    }
end

```

```

        replace `ll' = (binormal((`c12'-'Y1'),`eta'*(`c22'-'Y2)'),`rt')-binormal((`c11'-'
`Y1'),`eta'*(`c22'-'Y2)'),`rt')-binormal((`c12'-'Y1'),`eta'*(`c21'-'Y2)'),`rt')+binormal((`c11'-'
`Y1'),`eta'*(`c21'-'Y2)'),`rt') if m6==1
        replace `ll' = (binormal((`c12'-'Y1'),`eta'*(`c23'-'Y2)'),`rt')-binormal((`c11'-'
`Y1'),`eta'*(`c23'-'Y2)'),`rt')-binormal((`c12'-'Y1'),`eta'*(`c22'-'Y2)'),`rt')+binormal((`c11'-'
`Y1'),`eta'*(`c22'-'Y2)'),`rt') if m7==1
        replace `ll' = (binormal((`c12'-'Y1'),`eta'*(`c24'-'Y2)'),`rt')-binormal((`c11'-'
`Y1'),`eta'*(`c24'-'Y2)'),`rt')-binormal((`c12'-'Y1'),`eta'*(`c23'-'Y2)'),`rt')+binormal((`c11'-'
`Y1'),`eta'*(`c23'-'Y2)'),`rt') if m8==1
        replace `ll' = (normal((`c12'-'Y1'))-normal((`c11'-'Y1'))-binormal((`c12'-'
`Y1'),`eta'*(`c24'-'Y2)'),`rt')+binormal((`c11'-'Y1'),`eta'*(`c24'-'Y2)'),`rt') if m9==1

        replace `ll' = (binormal((`c13'-'Y1'),`eta'*(`c21'-'Y2)'),`rt')-binormal((`c12'-'
`Y1'),`eta'*(`c21'-'Y2)'),`rt') if m10==1
        replace `ll' = (binormal((`c13'-'Y1'),`eta'*(`c22'-'Y2)'),`rt')-binormal((`c12'-'
`Y1'),`eta'*(`c22'-'Y2)'),`rt')-binormal((`c13'-'Y1'),`eta'*(`c21'-'Y2)'),`rt')+binormal((`c12'-'
`Y1'),`eta'*(`c21'-'Y2)'),`rt') if m11==1
        replace `ll' = (binormal((`c13'-'Y1'),`eta'*(`c23'-'Y2)'),`rt')-binormal((`c12'-'
`Y1'),`eta'*(`c23'-'Y2)'),`rt')-binormal((`c13'-'Y1'),`eta'*(`c22'-'Y2)'),`rt')+binormal((`c12'-'
`Y1'),`eta'*(`c22'-'Y2)'),`rt') if m12==1
        replace `ll' = (binormal((`c13'-'Y1'),`eta'*(`c24'-'Y2)'),`rt')-binormal((`c12'-'
`Y1'),`eta'*(`c24'-'Y2)'),`rt')-binormal((`c13'-'Y1'),`eta'*(`c23'-'Y2)'),`rt')+binormal((`c12'-'
`Y1'),`eta'*(`c23'-'Y2)'),`rt') if m13==1
        replace `ll' = (normal((`c13'-'Y1'))-normal((`c12'-'Y1'))-binormal((`c13'-'
`Y1'),`eta'*(`c24'-'Y2)'),`rt')+binormal((`c12'-'Y1'),`eta'*(`c24'-'Y2)'),`rt') if m14==1

        replace `ll' = (binormal((`c14'-'Y1'),`eta'*(`c21'-'Y2)'),`rt')-binormal((`c13'-'
`Y1'),`eta'*(`c21'-'Y2)'),`rt') if m15==1
        replace `ll' = (binormal((`c14'-'Y1'),`eta'*(`c22'-'Y2)'),`rt')-binormal((`c13'-'
`Y1'),`eta'*(`c22'-'Y2)'),`rt')-binormal((`c14'-'Y1'),`eta'*(`c21'-'Y2)'),`rt')+binormal((`c13'-'
`Y1'),`eta'*(`c21'-'Y2)'),`rt') if m16==1
        replace `ll' = (binormal((`c14'-'Y1'),`eta'*(`c23'-'Y2)'),`rt')-binormal((`c13'-'
`Y1'),`eta'*(`c23'-'Y2)'),`rt')-binormal((`c14'-'Y1'),`eta'*(`c22'-'Y2)'),`rt')+binormal((`c13'-'
`Y1'),`eta'*(`c22'-'Y2)'),`rt') if m17==1
        replace `ll' = (binormal((`c14'-'Y1'),`eta'*(`c24'-'Y2)'),`rt')-binormal((`c13'-'
`Y1'),`eta'*(`c24'-'Y2)'),`rt')-binormal((`c14'-'Y1'),`eta'*(`c23'-'Y2)'),`rt')+binormal((`c13'-'
`Y1'),`eta'*(`c23'-'Y2)'),`rt') if m18==1
        replace `ll' = (normal((`c14'-'Y1'))-normal((`c13'-'Y1'))-binormal((`c14'-'
`Y1'),`eta'*(`c24'-'Y2)'),`rt')+binormal((`c13'-'Y1'),`eta'*(`c24'-'Y2)'),`rt') if m19==1

        replace `ll' = (normal(`eta'*(`c21'-'Y2'))- binormal((`c14'-'Y1'),`eta'*(`c21'-'
`Y2)'),`rt') if m20==1
        replace `ll' = (normal(`eta'*(`c22'-'Y2'))-binormal((`c14'-'Y1'),`eta'*(`c22'-'
`Y2)'),`rt')-normal(`eta'*(`c21'-'Y2'))+binormal((`c14'-'Y1'),`eta'*(`c21'-'Y2)'),`rt') if m21==1
        replace `ll' = (normal(`eta'*(`c23'-'Y2'))-binormal((`c14'-'Y1'),`eta'*(`c23'-'
`Y2)'),`rt')-normal(`eta'*(`c22'-'Y2'))+binormal((`c14'-'Y1'),`eta'*(`c22'-'Y2)'),`rt') if m22==1
        replace `ll' = (normal(`eta'*(`c24'-'Y2'))-binormal((`c14'-'Y1'),`eta'*(`c24'-'
`Y2)'),`rt')-normal(`eta'*(`c23'-'Y2'))+binormal((`c14'-'Y1'),`eta'*(`c24'-'Y2)'),`rt') if m23==1
        replace `ll' = (1-normal((`c14'-'Y1'))-normal(`eta'*(`c24'-'Y2'))+binormal((`c14'-'
`Y1'),`eta'*(`c24'-'Y2)'),`rt') if m24==1

        replace `L1'=(1/sqrt(2*asin(1)))*w`r'*`ll'
        replace `L2'=`L2'+`L1'
    }
}

quietly replace `lnf'=ln(`L2') if $ML_y1 ==1 & $ML_y2 == 1
quietly replace `lnf'=ln(`L2') if $ML_y1 ==1 & $ML_y2 == 2
quietly replace `lnf'=ln(`L2') if $ML_y1 ==1 & $ML_y2 == 3
quietly replace `lnf'=ln(`L2') if $ML_y1 ==1 & $ML_y2 == 4
quietly replace `lnf'=ln(`L2') if $ML_y1 ==1 & $ML_y2 == 5

```

```

quietly replace `lnf'=ln(`L2') if $ML_y1 ==2 & $ML_y2 == 1
quietly replace `lnf'=ln(`L2') if $ML_y1 ==2 & $ML_y2 == 2
quietly replace `lnf'=ln(`L2') if $ML_y1 ==2 & $ML_y2 == 3
quietly replace `lnf'=ln(`L2') if $ML_y1 ==2 & $ML_y2 == 4
quietly replace `lnf'=ln(`L2') if $ML_y1 ==2 & $ML_y2 == 5

quietly replace `lnf'=ln(`L2') if $ML_y1 ==3 & $ML_y2 == 1
quietly replace `lnf'=ln(`L2') if $ML_y1 ==3 & $ML_y2 == 2
quietly replace `lnf'=ln(`L2') if $ML_y1 ==3 & $ML_y2 == 3
quietly replace `lnf'=ln(`L2') if $ML_y1 ==3 & $ML_y2 == 4
quietly replace `lnf'=ln(`L2') if $ML_y1 ==3 & $ML_y2 == 5

quietly replace `lnf'=ln(`L2') if $ML_y1 ==4 & $ML_y2 == 1
quietly replace `lnf'=ln(`L2') if $ML_y1 ==4 & $ML_y2 == 2
quietly replace `lnf'=ln(`L2') if $ML_y1 ==4 & $ML_y2 == 3
quietly replace `lnf'=ln(`L2') if $ML_y1 ==4 & $ML_y2 == 4
quietly replace `lnf'=ln(`L2') if $ML_y1 ==4 & $ML_y2 == 5

quietly replace `lnf'=ln(`L2') if $ML_y1 ==5 & $ML_y2 == 1
quietly replace `lnf'=ln(`L2') if $ML_y1 ==5 & $ML_y2 == 2
quietly replace `lnf'=ln(`L2') if $ML_y1 ==5 & $ML_y2 == 3
quietly replace `lnf'=ln(`L2') if $ML_y1 ==5 & $ML_y2 == 4
quietly replace `lnf'=ln(`L2') if $ML_y1 ==5 & $ML_y2 == 5

exit

end

program define myprog2, rclass
syntax [, obs(integer 1) rho(real 0) mg(real 0) sg(real 0) ]

clear

set more off

*create a bivariate standard normal distribution with correlation 0.5, u1 mean 0, u2 mean 0, u1
sd 1, u2 sd 1:*
matrix C = (1, `rho'\ `rho', 1)
matrix m = (0,0)
drawnorm e1 e2, n(`obs') corr(C) means(m)

*generate covariates*
gen x1 = rnormal()
gen z1 = rnormal()

gen gamma = `mg'+`sg'*rnormal()

gen y1star =          + 1*x1 + 1*z1 + e1
gen y2star = gamma*y1star + 1*x1          + e2

gen y1 = 1 if y1star <- 1
replace y1=2 if y1star>-1&y1star<=0
replace y1=3 if y1star>0&y1star<=1
replace y1=4 if y1star>1&y1star<=2
replace y1=5 if y1star>2

gen y2 = 1 if y2star <-1
replace y2=2 if y2star>-1&y2star<=0
replace y2=3 if y2star>0&y2star<=1
replace y2=4 if y2star>1&y2star<=2
replace y2=5 if y2star>2

```

```
gen m0=(y1==1&y2==1)
gen m1=(y1==1&y2==2)
gen m2=(y1==1&y2==3)
gen m3=(y1==1&y2==4)
gen m4=(y1==1&y2==5)

gen m5=(y1==2&y2==1)
gen m6=(y1==2&y2==2)
gen m7=(y1==2&y2==3)
gen m8=(y1==2&y2==4)
gen m9=(y1==2&y2==5)

gen m10=(y1==3&y2==1)
gen m11=(y1==3&y2==2)
gen m12=(y1==3&y2==3)
gen m13=(y1==3&y2==4)
gen m14=(y1==3&y2==5)

gen m15=(y1==4&y2==1)
gen m16=(y1==4&y2==2)
gen m17=(y1==4&y2==3)
gen m18=(y1==4&y2==4)
gen m19=(y1==4&y2==5)

gen m20=(y1==5&y2==1)
gen m21=(y1==5&y2==2)
gen m22=(y1==5&y2==3)
gen m23=(y1==5&y2==4)
gen m24=(y1==5&y2==5)

gen double a1=.
replace a1=-2.453407083009012499e-01
gen double a2=.
replace a2=-1.234076215395323007e+00
gen double a3=.
replace a3=-2.254974002089275523e+00
gen double a4=.
replace a4=-3.347854567383216326e+00
gen double a5=.
replace a5=-4.603682449550744272e+00
gen double a6=.
replace a6=-7.374737285453943587e-01
gen double a7=.
replace a7=-1.738537712116586206e+00
gen double a8=.
replace a8=-2.788806058428130480e+00
gen double a9=.
replace a9=-3.944764040115625210e+00
gen double a10=.
replace a10=-5.387480890011232861e+00
gen double a11=.
replace a11=2.453407083009012499e-01
gen double a12=.
replace a12=1.234076215395323007e+00
gen double a13=.
replace a13=2.254974002089275523e+00
gen double a14=.
replace a14=3.347854567383216326e+00
gen double a15=.
replace a15=4.603682449550744272e+00
gen double a16=.
```

```
replace a16=7.374737285453943587e-01
gen double a17=.
replace a17=1.738537712116586206e+00
gen double a18=.
replace a18=2.788806058428130480e+00
gen double a19=.
replace a19=3.944764040115625210e+00
gen double a20=.
replace a20=5.387480890011232861e+00
```

```
gen double w1=.
replace w1=4.622436696006100896e-01
gen double w2=.
replace w2=1.090172060200233200e-01
gen double w3=.
replace w3=3.243773342237861832e-03
gen double w4=.
replace w4=7.802556478532063693e-06
gen double w5=.
replace w5=4.399340992273180553e-10
gen double w6=.
replace w6=2.866755053628341297e-01
gen double w7=.
replace w7=2.481052088746361088e-02
gen double w8=.
replace w8=2.283386360163539672e-04
gen double w9=.
replace w9=1.086069370769281693e-07
gen double w10=.
replace w10=2.229393645534151292e-13
gen double w11=.
replace w11=4.622436696006100896e-01
gen double w12=.
replace w12=1.090172060200233200e-01
gen double w13=.
replace w13=3.243773342237861832e-03
gen double w14=.
replace w14=7.802556478532063693e-06
gen double w15=.
replace w15=4.399340992273180553e-10
gen double w16=.
replace w16=2.866755053628341297e-01
gen double w17=.
replace w17=2.481052088746361088e-02
gen double w18=.
replace w18=2.283386360163539672e-04
gen double w19=.
replace w19=1.086069370769281693e-07
gen double w20=.
replace w20=2.229393645534151292e-13
```

```
global draws "20"
cap drop l logl r
gen l=0
gen logl = 0
gen r = 0
```

```
*Starting values
    set more off
    set seed 12347
    tab y1
```



```

tab y2

mat drop _all
oprobit y1 x1 z1
mat b1 = e(b)
mat coleq b1 = y1
oprobit y2 y1 x1
mat c2 = e(b)
mat list c2
local last = colsof(c2)
mat b2 = c2[1,2.. `last']
mat coleq b2 = exam
mat b3 = 1,1,1
mat coleq b3 = mg/sg/rho
mat start = b1, b2, b3
mat list start

*model
xi: ml model lf sm_ml_lf (y1 = x1 z1, noconstant ) (c11:) (c12:) (c13:) (c14:) (y2 = x1 ,
noconstant) (c21:) (c22:) (c23:) (c24:) (mg:) (sg:) (r:) , technique(bhhh)
ml init start, copy
ml max, iterate(200) difficult trace

estimates store a

nlcom (meang: ([mg]_b[_cons]))
nlcom (sigmag: exp([sg]_b[_cons]))
nlcom (rho: tanh([r]_b[_cons]))

gen mg = ([mg]_b[_cons])
gen sg = exp([sg]_b[_cons])
gen rho= tanh([r]_b[_cons])

gen rmsemg = sqrt((mg-(`mg'))^2)
gen rmsesg = sqrt((sg-(`sg'))^2)
gen rmserho = sqrt((rho-(`rho'))^2)

end

simulate mg sg rho rmsemg rmsesg rmserho, reps(1000) : myprog2, obs(1000) rho(-0.9) mg(1) sg(1)
su
simulate mg sg rho rmsemg rmsesg rmserho, reps(1000) : myprog2, obs(1000) rho(-0.5) mg(1) sg(1)
su
simulate mg sg rho rmsemg rmsesg rmserho, reps(1000) : myprog2, obs(1000) rho(0) mg(1) sg(1)
su
simulate mg sg rho rmsemg rmsesg rmserho, reps(1000) : myprog2, obs(1000) rho(0.5) mg(1) sg(1)
su
simulate mg sg rho rmsemg rmsesg rmserho, reps(1000) : myprog2, obs(1000) rho(0.9) mg(1) sg(1)
su
*****

```