OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

WW FACULTY OF ECONOMICS AND MANAGEMENT

Simulated Multivariate Random Effects Probit Models for Unbalanced Panels

Alexander Plum

2013 German Stata Users Group Meeting

June 7, 2013

Overview

Introduction

Random Effects Model

Illustration
- Simulated Multivariate Random Effects Probit Model for Unbalanced Panels
- Robustness check I
- Robustness check II

Extending to Autocorrelated Errors

Introduction

Dynamic models:

- Past outcome (y_{it-1}) $\Rightarrow$ current outcome (y_{it})
  - Stigmatization of unemployment (Arulampalam et al., 2000)
  - Stepping-stone effect of low-paid employment (Stewart, 2007)

Introduction

Dynamic models:

- Past outcome (y_{it-1}) $\Rightarrow$ current outcome (y_{it})
  - Stigmatization of unemployment (Arulampalam et al., 2000)
  - Stepping-stone effect of low-paid employment (Stewart, 2007)
- Time-invariant error term (Heckman 1981a)

Introduction

Dynamic models:

- Past outcome (y_{it-1}) $\Rightarrow$ current outcome (y_{it})
    - Stigmatization of unemployment (Arulampalam et al., 2000)
    - Stepping-stone effect of low-paid employment (Stewart, 2007)
- Time-invariant error term (Heckman 1981a)
- *Initial condition problem* (Heckman 1981b)

Introduction

Several Stata commands exist:

- **redprob** or **redpace** (Stewart 2006a,b)
- Based on (adaptive) Gaussian-Hermite quadratures or on Maximum Simulated Likelihood (*MSL*)
- Restricted to balanced panels

Introduction

Simulated Multivariate Random Effects Probit Model:

Introduction

Simulated Multivariate Random Effects Probit Model:

1. Unbalanced panels

Introduction

Simulated Multivariate Random Effects Probit Model:

1. Unbalanced panels
2. Estimator can easily be adjusted, e.g. to allow for autocorrelated errors

Introduction

Simulated Multivariate Random Effects Probit Model:

1. Unbalanced panels
2. Estimator can easily be adjusted, e.g. to allow for autocorrelated errors
3. High accuracy

Introduction

Simulated Multivariate Random Effects Probit Model:

1. Unbalanced panels
2. Estimator can easily be adjusted, e.g. to allow for autocorrelated errors
3. High accuracy
4. Lower computational time

Random Effects Model

The latent variable y^*_{it} is specified for $t \geq 2, \ldots, T$ by:

$$y^*_{it} = \gamma y^*_{it-1} + x'_{it}\beta + \alpha_i + u_{it}. \tag{1}$$

The observed binary outcome variable is defined as:

$$y_{it} = \begin{cases} 1 & \text{if } y^*_{it} > 0, \\ 0 & \text{else.} \end{cases} \tag{2}$$

The composite error term is $\nu_{it} = \alpha_i + u_{it}$ with $u_{it} \sim N(0, 1)$ and $\alpha_i \sim N(0, \sigma^2_\alpha)$. The composite error term takes the following equi-correlation structure over time (with $t \neq s$):

$$corr(\nu_{it}, \nu_{is}) = \sigma^2_\alpha. \tag{3}$$

Random Effects Model

Following the approach of Heckman (1981b) for the initial condition problem:

$$y_{i1}^{*} = z_{i1}'\pi + \epsilon_i, \tag{4}$$

Correlation of the error term:

$$\epsilon_i = \theta\alpha_i + u_{i1}. \tag{5}$$

The correlation of the composite error term between the initial period and the subsequent ones is:

$$corr(\epsilon_i, \nu_{it}) = \theta\sigma_{\alpha}^{2}, \tag{6}$$

Random Effects Model

The variance-covariance matrix takes following form:

$$\Omega = \begin{pmatrix} \theta^2\sigma_\alpha^2 + 1 & & & & \\ \theta\sigma_\alpha^2 & \sigma_\alpha^2 + 1 & & & \\ \theta\sigma_\alpha^2 & \sigma_\alpha^2 & \sigma_\alpha^2 + 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \theta\sigma_\alpha^2 & \sigma_\alpha^2 & \sigma_\alpha^2 & \dots & \sigma_\alpha^2 + 1 \end{pmatrix}. \quad (7)$$

Random Effects Model

The likelihood-contribution of each individual is:

$$\Phi_{iT} = (k_{i1} z'_{i1}\pi, k_{i2} x'_{i2}\beta, \ldots, k_{iT} x'_{iT}\beta,$$
$$k_{i1}k_{i2}\Omega_{2,1}, k_{i1}k_{i3}\Omega_{3,1}, \ldots, k_{iT-1}k_{iT}\Omega_{T,T-1}). \tag{8}$$

There are T sign variables k_{it}, with:

$$k_{it} = \begin{cases} 1 & \text{if } y_{it} = 1, \\ -1 & \text{else.} \end{cases} \tag{9}$$

Random Effects Model

The log likelihood to be maximized is the sum of the individual log likelihood contributions:

$$\ln L = \ln \sum_{i=1}^{N} \Phi_{iT}(\mu; \Omega), \tag{10}$$

Note: $\mu = (k_{i1} z'_{i1} \pi, \ldots, k_{iT} x'_{iT} \beta), \Omega = (k_{i1} k_{i2} \Omega_{2,1}, \ldots, k_{iT-1} k_{iT} \Omega_{T,T-1})$.

Random Effects Model

- Multivariate normal probability functions of order T required

Random Effects Model

- Multivariate normal probability functions of order T required
- In Stata, only the bivariate normal distribution function exists

Random Effects Model

- Multivariate normal probability functions of order T required
- In Stata, only the bivariate normal distribution function exists
- Simulated multivariate normal probabilities are derived by the command `mvnp`

Random Effects Model

- Multivariate normal probability functions of order T required
- In Stata, only the bivariate normal distribution function exists
- Simulated multivariate normal probabilities are derived by the command `mvnp`
- Using Halton draws, which are generated with `mdraws`

Random Effects Model

- Multivariate normal probability functions of order T required
- In Stata, only the bivariate normal distribution function exists
- Simulated multivariate normal probabilities are derived by the command `mvnp`
- Using Halton draws, which are generated with `mdraws`
- The total number of generated Halton draws is R and with each draw $r \in \{1, \ldots, R\}$ multivariate normal probabilities are simulated and the average of these simulations is derived.

Random Effects Model

- Multivariate normal probability functions of order T required
- In Stata, only the bivariate normal distribution function exists
- Simulated multivariate normal probabilities are derived by the command `mvnp`
- Using Halton draws, which are generated with `mdraws`
- The total number of generated Halton draws is R and with each draw $r \in \{1, \ldots, R\}$ multivariate normal probabilities are simulated and the average of these simulations is derived.

Hence, the logarithm of the simulated likelihood is:

$$\ln SL = \ln \frac{1}{R} \sum_{r=1}^{R} \sum_{i=1}^{N} \Phi_{iT}^{r}(\mu; \Omega). \tag{11}$$

Illustration

Creating an artificial data set:

Illustration

Creating an artificial data set:

- 1000 individuals, 5 time periods

Illustration

Creating an artificial data set:

- 1000 individuals, 5 time periods
- Time-invariant error term (`alpha`), explanatory (`x1`,`x2`,`x3`) and instrumental variables (`Instrument`), idiosyncratic shock (`u_i`) and a variable called `Random`

Illustration

Creating an artificial data set:

- 1000 individuals, 5 time periods
- Time-invariant error term (`alpha`), explanatory (`x1`,`x2`,`x3`) and instrumental variables (`Instrument`), idiosyncratic shock (`u_i`) and a variable called `Random`
- Time-invariant error term has a normalization of $\sim N(0, 2)$, all other variables are standard normal distributed, i.e. $\sim N(0, 1)$

Illustration

Creating an artificial data set:

- 1000 individuals, 5 time periods
- Time-invariant error term (`alpha`), explanatory (`x1`,`x2`,`x3`) and instrumental variables (`Instrument`), idiosyncratic shock (`u_i`) and a variable called `Random`
- Time-invariant error term has a normalization of $\sim N(0, 2)$, all other variables are standard normal distributed, i.e. $\sim N(0, 1)$
- The variable `Random` is a temporary identifier which helps to construct an unbalanced panel

Illustration

```
set obs 1000
gen id=_n
expand 5
bys id:  gen tper=_n
```

Illustration

```
set obs 1000
gen id=_n
expand 5
bys id:  gen tper=_n

matrix m = (0,0,0,0,0,0,0)
matrix sd = (sqrt(2),1,1,1,1,1,1)
drawnorm alpha Instrument x1 x2 x3 u_i Random,
n(5000) means(m) sds(sd) seed(987654321)
replace Random=normal(Random)
```

Illustration

```
sort id tper
by id:  replace alpha=alpha[1]
by id:  replace Random=Random[1]
drop if tper==5 & Random>.85
drop if tper>=4 & Random<.10
bys id (tper):  gen nwave=_N
```

Illustration

The latent variable y^* is constructed in the following manner:

$$y^*_{i1} = 0.7 + 0.35x_1 + 0.66x_2 + 0.25x_3 + 1.5x_{\text{Instrument}} + \theta\alpha_i + u_{i1},$$

where $x_{\text{Instrument}}$ is an instrumental variable which will only have an effect on the outcome of the initial period and not on the subsequent ones. For the initial period it is assumed that θ takes on the value 1. For the subsequent periods $t = 2, \ldots, 5$ the following relationship is defined:

$$y^*_{it} = 0.3 + 0.46y_{t-1} + 0.25x_1 + 0.75x_2 + 0.55x_3 + \alpha_i + u_{it}.$$

The observable variable y_{it} becomes 1 if $y^*_{it} > 0$ and 0 else. Furthermore, the variable `ylag` is generated which takes the value of the outcome variable of the previous period.

Illustration

```
sort id (tper)
local theta=1
by id:  gen ystar=.35*x1 + .66*x2 + .25*x3 +
1.5*Instrument + .7 + ‘theta’*alpha + u_i if _n==1
by id:  gen y=cond(ystar>0,1,0) if _n==1
```

Illustration

```
sort id (tper)
local theta=1
by id:  gen ystar=.35*x1 + .66*x2 + .25*x3 +
1.5*Instrument + .7 + ‘theta’*alpha + u_i if _n==1
by id:  gen y=cond(ystar>0,1,0) if _n==1

sort id (tper)
forvalues i=2/5{
by id:  replace ystar =.25*x1 + .75*x2 + .55*x3 +
.46*y[_n-1] + .35 + alpha + u_i if _n==‘i’
by id:  replace y=cond(ystar>0,1,0) if _n==‘i’
}
sort id (tper)
by id:  gen ylag=cond(_n>1,y[_n-1],.)
```

Illustration

```
matrix p=(2,3,5,7,11)
mdraws, neq(5) draws(100) prefix(z) primes(p)
burn(15)
```

Created 100 Halton draws per equation for 5 dimensions. Number of initial draws dropped per dimension = 15 . Primes used: 2 3 5 7 11

Illustration

```
matrix p=(2,3,5,7,11)
mdraws, neq(5) draws(100) prefix(z) primes(p)
burn(15)
```

Created 100 Halton draws per equation for 5 dimensions. Number of initial draws dropped per dimension = 15 . Primes used: 2 3 5 7 11

```
global dr = r(n_draws)
global T_max=5
global T_min=3
```

Stata Syntax

```
cap prog drop mpheckman_d0
program define mpheckman_d0
args todo b lnf
tempname sigma theta
tempvar beta pi lnsigma lntheta T fi fi6 fi5 fi4 fi3 FF
mleval `beta' = `b', eq(1)
mleval `pi' = `b', eq(2)
mleval `lnsigma' = `b', eq(3) scalar
mleval `lntheta' = `b', eq(4) scalar
```

Stata Syntax

```
cap prog drop mpheckman_d0
program define mpheckman_d0
args todo b lnf
tempname sigma theta
tempvar beta pi lnsigma lntheta T fi fi6 fi5 fi4 fi3 FF
mleval `beta' = `b', eq(1)
mleval `pi' = `b', eq(2)
mleval `lnsigma' = `b', eq(3) scalar
mleval `lntheta' = `b', eq(4) scalar

 scalar `sigma'=(exp(`lnsigma'))^2
scalar `theta'=exp(`lntheta')
```

Stata Syntax

```
cap prog drop mpheckman_d0
program define mpheckman_d0
args todo b lnf
tempname sigma theta
tempvar beta pi lnsigma lntheta T fi fi6 fi5 fi4 fi3 FF
mleval ‘beta’ = ‘b’, eq(1)
mleval ‘pi’ = ‘b’, eq(2)
mleval ‘lnsigma’ = ‘b’, eq(3) scalar
mleval ‘lntheta’ = ‘b’, eq(4) scalar

 scalar ‘sigma’=(exp(‘lnsigma’))^2
scalar ‘theta’=exp(‘lntheta’)

 qui:{
by idcode:  gen double ‘T’ = (_n == _N)
sort idcode (year)
tempvar k1 zb1
by idcode:  gen double ‘k1’ = (2*$ML_y1[1]) - 1
by idcode:  gen double ‘zb1’ = ‘pi’[1]
forvalues r = 2/$T_max {
tempvar k‘r’ xb‘r’
by idcode:  gen double ‘k‘r’’ = (2*$ML_y1[‘r’]) - 1
by idcode:  gen double ‘xb‘r’’ = ‘beta’[‘r’]
}
```

Stata Syntax

```
forvalues s=$T_min/$T_max{
tempname V`s' C`s'
}
mat `V$T_max'=I($T_max)*(`sigma'+1)
mat `V$T_max'[1,1]=(`theta'^2)*`sigma'+1
```

Stata Syntax

```
forvalues s=$T_min/$T_max{
tempname V‘s’ C‘s’
}
mat ‘V$T_max’=I($T_max)*(‘sigma’+1)
mat ‘V$T_max’[1,1]=(‘theta’^2)*‘sigma’+1

 forvalues row=2/$T_max{
mat ‘V$T_max’[‘row’,1] = (‘theta’*‘sigma’)
mat ‘V$T_max’[1,‘row’] = ‘V$T_max’[‘row’,1]
local s = ‘row’-1
forvalues col=2/‘s’{
mat ‘V$T_max’[‘row’,‘col’] = ‘sigma’
mat ‘V$T_max’[‘col’,‘row’] = ‘V$T_max’[‘row’,‘col’]
}
}
```

Stata Syntax

```
forvalues s=$T_min/$T_max{
tempname V‘s’ C‘s’
}
mat ‘V$T_max’=I($T_max)*(‘sigma’+1)
mat ‘V$T_max’[1,1]=(‘theta’^2)*‘sigma’+1

 forvalues row=2/$T_max{
mat ‘V$T_max’[‘row’,1] = (‘theta’*‘sigma’)
mat ‘V$T_max’[1,‘row’] = ‘V$T_max’[‘row’,1]
local s = ‘row’-1
forvalues col=2/‘s’{
mat ‘V$T_max’[‘row’,‘col’] = ‘sigma’
mat ‘V$T_max’[‘col’,‘row’] = ‘V$T_max’[‘row’,‘col’]
}
}

 forvalues r = $T_min/$T_max{
mat ‘V‘r’’ = ‘V$T_max’[1..‘r’,1..‘r’]
mat ‘C‘r’’ = cholesky(‘V‘r’’)
}
```

Stata Syntax

```
egen double ‘fi5’ = mvnp(‘zb1’ ‘xb2’ ‘xb3’ ‘xb4’ ‘xb5’) if nwave==5, /*
*/ chol(‘C5’) dr($dr) prefix(z) signs(‘k1’ ‘k2’ ‘k3’ ‘k4’ ‘k5’) adoonly
egen double ‘fi4’ = mvnp(‘zb1’ ‘xb2’ ‘xb3’ ‘xb4’) if nwave==4, /*
*/ chol(‘C4’) dr($dr) prefix(z) signs(‘k1’ ‘k2’ ‘k3’ ‘k4’) adoonly
egen double ‘fi3’ = mvnp(‘zb1’ ‘xb2’ ‘xb3’) if nwave==3, /*
*/ chol(‘C3’) dr($dr) prefix(z) signs(‘k1’ ‘k2’ ‘k3’) adoonly
```

Stata Syntax

```
egen double ‘fi5’ = mvnp(‘zb1’ ‘xb2’ ‘xb3’ ‘xb4’ ‘xb5’) if nwave==5, /*
*/ chol(‘C5’) dr($dr) prefix(z) signs(‘k1’ ‘k2’ ‘k3’ ‘k4’ ‘k5’) adoonly
egen double ‘fi4’ = mvnp(‘zb1’ ‘xb2’ ‘xb3’ ‘xb4’) if nwave==4, /*
*/ chol(‘C4’) dr($dr) prefix(z) signs(‘k1’ ‘k2’ ‘k3’ ‘k4’) adoonly
egen double ‘fi3’ = mvnp(‘zb1’ ‘xb2’ ‘xb3’) if nwave==3, /*
*/ chol(‘C3’) dr($dr) prefix(z) signs(‘k1’ ‘k2’ ‘k3’) adoonly

 gen double ‘fi’=cond(nwave==5,‘fi5’,cond(nwave==4,‘fi4’,‘fi3’))
gen double ‘FF’ = cond(!‘T’,0,ln(‘fi’))
}
mlsum ‘lnf’ = ‘FF’ if ‘T’
if (‘todo’==0 | ‘lnf’>=.)  exit

end
```

Initial values

```
qui: probit y ylag x1 x2 x3 if tper> 1
matrix b0=e(b)
qui: probit y x1 x2 x3 Instrument if tper==1
matrix b1=e(b)
matrix b12 = (-.5,-.5)
matrix b0 = (b0 , b1 , b12)
```

Stata output

```
ml model d0 mpheckman_d0 (y: y = ylag x1 x2 x3) (Init_Period: y = x1
x2 x3 Instrument) /lnsigma /lntheta, title(Multivariate RE Probit, $dr
Halton draws) missing

ml init b0, copy

ml max
```

(*output omitted*)

Stata output

Multivariate RE Probit, 100 Halton draws

Number of obs = 4689
Wald chi2(4) = 460.84
Log likelihood = -2099.9876
Prob > chi2 = 0.0000

	Coef.	Std. Err.	z	P> |z|	[95% Conf.	Interval]
y						
ylag	.4598806	.0813738	5.65	0.000	.300391	.6193703
x1	.3074512	.0357443	8.60	0.000	.2373936	.3775087
x2	.7470318	.0427175	17.49	0.000	.663307	.8307565
x3	.5663907	.0390205	14.52	0.000	.489912	.6428694
_cons	.3250167	.0821635	3.96	0.000	.1639792	.4860542
Init_Period						
x1	.3800084	.0733688	5.18	0.000	.2362083	.5238086
x2	.7001715	.0858233	8.16	0.000	.5319609	.868382
x3	.3487215	.0737431	4.73	0.000	.2041876	.4932553
Instrument	1.518743	.1419662	10.70	0.000	1.240495	1.796992
_cons	.705813	.0944883	7.47	0.000	.5206193	.8910066
lnsigma						
_cons	.3597355	.0681636	5.28	0.000	.2261373	.4933338
lntheta						
_cons	-.0438069	.1375578	-0.32	0.750	-.3134153	.2258016

Stata output

Transforming of `lnsigma` and `lntheta` to derive σ_α^2 and θ:

```
_diparm lnsigma, function((exp(@))^2) deriv(2*(exp(@))*(exp(@)))
label("‘Sigma2") prob

_diparm lntheta, function(exp(@)) deriv(exp(@)) label("Theta") prob
```

Sigma^2	2.053347	.2799271	7.34	0.000	1.571884	2.682281
Theta	.9571388	.131662	7.27	0.000	.7309463	1.253327

Robustness check I

Robustness check:

- Applying different sets of primes; picked randomly in the range between $2, \ldots, 97$
- 10 estimations run

$\Rightarrow$ Results only differ slightly!

Robustness check II

Robustness check:

- Results compared with those of the command `redpace`
- Identical data set created, but balanced this time
- Estimations are run on the basis of 20, 50 and 100 draws (Halton draws and pseudo-random numbers)
- Indicator for efficiency: log-likelihood and computational time

Robustness check II

Results:

Robustness check II

Results:

1. When 100 draws applied all estimators derive similar coefficients and log-likelihood

Robustness check II

Results:

1. When 100 draws applied all estimators derive similar coefficients and log-likelihood
2. Computational time lower in the multivariate random effects probit model (between ∼28% and ∼38%)

Robustness check II

Results:

1. When 100 draws applied all estimators derive similar coefficients and log-likelihood
2. Computational time lower in the multivariate random effects probit model (between ∼28% and ∼38%)
3. When 20 Halton draws are applied, multivariate random effects probit model is more accurate

Extending to Autocorrelated Errors

Assumption by now is that the idiosyncratic shock is autocorrelated so that it follows a *AR(1)*-process:

$$u_{it} = \delta u_{it-1} + \epsilon_{it}.$$

Extending to Autocorrelated Errors

Assumption by now is that the idiosyncratic shock is autocorrelated so that it follows a *AR(1)*-process:

$$u_{it} = \delta u_{it-1} + \epsilon_{it}.$$

The generalized variance-covariance matrix takes on following form:

$$\Omega = \begin{pmatrix} \theta^2\sigma_\alpha^2 + 1 & & & & & \\ \theta\sigma_\alpha^2 + \delta & \theta^2\sigma_\alpha^2 + 1 & & & & \\ \theta\sigma_\alpha^2 + \delta^2 & \sigma_\alpha^2 + \delta & \theta^2\sigma_\alpha^2 + 1 & & & \\ \theta\sigma_\alpha^2 + \delta^3 & \sigma_\alpha^2 + \delta^2 & \sigma_\alpha^2 + \delta & \theta^2\sigma_\alpha^2 + 1 & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ \theta\sigma_\alpha^2 + \delta^{T-1} & \sigma_\alpha^2 + \delta^{T-2} & \sigma_\alpha^2 + \delta^{T-3} & \sigma_\alpha^2 + \delta^{T-4} & \dots & \theta^2\sigma_\alpha^2 + 1 \end{pmatrix}$$

Extending to Autocorrelated Errors

Adjustments:

Extending to Autocorrelated Errors

Adjustments:

- Introducing the parameter ρ, which refers to the autocorrelated error term

Extending to Autocorrelated Errors

Adjustments:

- Introducing the parameter ρ, which refers to the autocorrelated error term
- Parameter ρ will be integrated into the Stata syntax as the inverse hyperbolic tangent of ρ

Extending to Autocorrelated Errors

Adjustments:

- Introducing the parameter ρ, which refers to the autocorrelated error term
- Parameter ρ will be integrated into the Stata syntax as the inverse hyperbolic tangent of ρ
- The variance-covariance matrix must be adjusted according to the adjusted Ω

Extending to Autocorrelated Errors

Findings:

Extending to Autocorrelated Errors

Findings:

- The findings go along with those of the `redpace` command, especially when 500 pseudo-random numbers are applied

Extending to Autocorrelated Errors

Findings:

- The findings go along with those of the `redpace` command, especially when 500 pseudo-random numbers are applied
- The log likelihood of the multivariate random effects probit model with autocorrelated errors only changes slightly when using 100 instead of 50 Halton quasi-random numbers

Extending to Autocorrelated Errors

Findings:

- The findings go along with those of the `redpace` command, especially when 500 pseudo-random numbers are applied
- The log likelihood of the multivariate random effects probit model with autocorrelated errors only changes slightly when using 100 instead of 50 Halton quasi-random numbers
- Accuracy can already be found for a low level of Halton draws and computational time can be saved when a multivariate random effects probit model is applied

Thank you for your attention!!!

References:

- Arulampalam, W. 1999. A note on estimated coefficients in random effects probit models. *Oxford Bulletin of Economics and Statistics* 76: 598-606.
- Arulampalam, W., A. Booth and M. Taylor. 2000. Unemployment persistence. *Oxford Economic Papers* 52: 24-50.
- Cappellari, L., and S.P. Jenkins 2006. Calculation of multivariate normal probabilities by simulation, with applications to maximum simulated likelihood estimation. *Stata Journal* 6: 156-189.
- Heckman, J.J. 1981a. *Heterogeneity and State Dependence*, in S. Rosen (ed.). Studies in Labor Market. Chicago: University of Chicago Press (for NBER).
- Heckman, J.J. 1981b. *The incidental parameters problem and the problem of initial conditions in estimating a discrete time - discrete data stochastic process*, in C.F. Manski and D. McFadden (eds.). Structural Analysis of Discrete Data with Econometric Applications. Cambridge: MIT Press.
- Stewart, M.B. 2006a. Maximum Simulated Likelihood Estimation of Random Effects Dynamic Probit Models with Autocorrelated Errors. *Stata Journal* 6: 256-272.
- Stewart, M.B. 2006b. Heckman estimator of the random effects dynamic probit model. From: `http://www2.warwick.ac.uk/fac/soc/economics/staff/academic/stewart/stata`
- Stewart, M.B. 2007. The Interrelated Dynamics of Unemployment and Low-Wage Employment. *Journal of Applied Econometrics* 22: 511-531.