GMM with first-step residuals:

A recipe for control-function S.E.s

Enrique Pinzón

StataCorp LLC

September 10, 2020
"London"

The plan

- Use `gmm` to get standard errors for control function type estimators
  - linear cross-sectional model
  - fractional (binary) outcome cross-sectional model
  - exponential mean (Poisson) panel-data model
- Control function estimates imply:
  - A test for endogeneity
  - A structural function interpretation of effects
- It is common to use the bootstrap
- Excuse to show you some `gmm` Jujutsu
- Discuss some estimation and postestimation considerations

Model I: Linear model

$$
\begin{aligned}
y &= X_1\beta_1 + X_2\beta_2 + \varepsilon \\
E\left(X_1'\varepsilon\right) &= \mathbf{0} \\
E\left(X_2'\varepsilon\right) &\neq \mathbf{0}
\end{aligned}
$$

- Different estimators arise from the following:

$$
\begin{aligned}
E\left(Z'\varepsilon\right) &= 0 && \text{Instrumental variables} \\
X_2 &= Z\Pi + \nu && \text{Two stage least squares (TSLS)} \\
E\left(Z'\nu\right) &= 0
\end{aligned}
$$

- Control function approaches additionally assume

$$
\begin{aligned}
\varepsilon &= \rho\nu + \epsilon \\
y &= X_1\beta_1 + X_2\beta_2 + \rho\nu + \epsilon
\end{aligned}
$$

Model I: Linear model

$$
\begin{aligned}
y &= X_1\beta_1 + X_2\beta_2 + \varepsilon \\
E\left(X_1'\varepsilon\right) &= \mathbf{0} \\
E\left(X_2'\varepsilon\right) &\neq \mathbf{0}
\end{aligned}
$$

- Different estimators arise from the following:

$$
\begin{aligned}
E\left(Z'\varepsilon\right) &= 0 && \text{Instrumental variables} \\
X_2 &= Z\Pi + \nu && \text{Two stage least squares (TSLS)} \\
E\left(Z'\nu\right) &= 0
\end{aligned}
$$

- Control function approaches additionally assume

$$
\begin{aligned}
\varepsilon &= \rho\nu + \epsilon \\
y &= X_1\beta_1 + X_2\beta_2 + \rho\nu + \epsilon
\end{aligned}
$$

Model I: Linear model

$$
\begin{aligned}
y &= X_1\beta_1 + X_2\beta_2 + \varepsilon \\
E\left(X_1'\varepsilon\right) &= \mathbf{0} \\
E\left(X_2'\varepsilon\right) &\neq \mathbf{0}
\end{aligned}
$$

- Different estimators arise from the following:

$$
\begin{aligned}
E\left(Z'\varepsilon\right) &= 0 && \text{Instrumental variables} \\
X_2 &= Z\Pi + \nu && \text{Two stage least squares (TSLS)} \\
E\left(Z'\nu\right) &= 0
\end{aligned}
$$

- Control function approaches additionally assume

$$
\begin{aligned}
\varepsilon &= \rho\nu + \epsilon \\
y &= X_1\beta_1 + X_2\beta_2 + \rho\nu + \epsilon
\end{aligned}
$$

Model I: Linear model

$$
\begin{aligned}
y &= X_1\beta_1 + X_2\beta_2 + \varepsilon \\
E\left(X_1'\varepsilon\right) &= \mathbf{0} \\
E\left(X_2'\varepsilon\right) &\neq \mathbf{0}
\end{aligned}
$$

- Different estimators arise from the following:

$$
\begin{aligned}
E\left(Z'\varepsilon\right) &= 0 && \text{Instrumental variables} \\
X_2 &= Z\Pi + \nu && \text{Two stage least squares (TSLS)} \\
E\left(Z'\nu\right) &= 0
\end{aligned}
$$

- Control function approaches additionally assume

$$
\begin{aligned}
\varepsilon &= \rho\nu + \epsilon \\
y &= X_1\beta_1 + X_2\beta_2 + \rho\nu + \epsilon
\end{aligned}
$$

TSLS and Control function

TSLS:

1. Regression of X_2 on Z, get part of X_2 without endogeneity, $\widehat{X}_2$
   - We get $\widehat{X}_2 = Z\left(Z'Z\right)^{-1}Z'X_2 = P_zX_2$
2. Regression of y on X_1 and $\widehat{X}_2$
   - We get $\widehat{\beta} = (X_2'P_zX_2)^{-1}X_2P_zy$

CONTROL FUNCTION (CF):

1. Get residuals from Regression of X_2 on Z
   - $\widehat{\nu} = X_2 - \widehat{X_2} = I - P_zX_2$
2. Regress y on X_1, X_2, and $\widehat{\nu}$

- We need to address uncertainty in estimation of ν

TSLS and Control function

TSLS:

1. Regression of X_2 on Z, get part of X_2 without endogeneity, $\widehat{X}_2$
   - We get $\widehat{X}_2 = Z\left(Z'Z\right)^{-1} Z'X_2 = P_z X_2$
2. Regression of y on X_1 and $\widehat{X}_2$
   - We get $\widehat{\beta} = (X_2' P_z X_2)^{-1} X_2 P_z y$

CONTROL FUNCTION (CF):

1. Get residuals from Regression of X_2 on Z
   - $\widehat{\nu} = X_2 - \widehat{X_2} = I - P_z X_2$
2. Regress y on X_1, X_2, and $\widehat{\nu}$

- We need to address uncertainty in estimation of ν

TSLS and Control function

TSLS:

1. Regression of X_2 on Z, get part of X_2 without endogeneity, $\widehat{X}_2$
   - We get $\widehat{X}_2 = Z\left(Z'Z\right)^{-1} Z' X_2 = P_z X_2$
2. Regression of y on X_1 and $\widehat{X}_2$
   - We get $\widehat{\beta} = (X_2' P_z X_2)^{-1} X_2 P_z y$

CONTROL FUNCTION (CF):

1. Get residuals from Regression of X_2 on Z
   - $\widehat{\nu} = X_2 - \widehat{X_2} = I - P_z X_2$
2. Regress y on X_1, X_2, and $\widehat{\nu}$

- We need to address uncertainty in estimation of ν

GMM for CF

- GMM needs specification $E\{W'e\} = \mathbf{0}$ or $E\{W'e(\theta)\} = \mathbf{0}$
- In gmm W are exogenous and specified as options
- $e(\theta)$ change at each iteration as θ converges to it's minimum
- The control function approach does not quite fit into gmm's framework. Let $W(\Pi) = [X, X_2 - Z\Pi]$

$$\begin{aligned} E\{Z'(X_2 - Z\Pi)\} &= \mathbf{0} \\ E\{W(\Pi)'[y - X_1\beta_1 + X_2\beta + \rho(X_2 - Z\Pi)]\} &= \mathbf{0} \end{aligned}$$

- Estimating $\widehat{\nu}$ and feeding it to gmm will give incorrect standard errors

GMM for CF

- GMM needs specification $E\{W'e\} = \mathbf{0}$ or $E\{W'e(\theta)\} = \mathbf{0}$
- In gmm W are exogenous and specified as options
- $e(\theta)$ change at each iteration as θ converges to it's minimum
- The control function approach does not quite fit into gmm's framework. Let $W(\Pi) = [X, X_2 - Z\Pi]$

$$\begin{aligned} E\{Z'(X_2 - Z\Pi)\} &= \mathbf{0} \\ E\{W(\Pi)'[y - X_1\beta_1 + X_2\beta + \rho(X_2 - Z\Pi)]\} &= \mathbf{0} \end{aligned}$$

- Estimating $\widehat{\nu}$ and feeding it to gmm will give incorrect standard errors

GMM for CF

- GMM needs specification $E\left\{W'e\right\} = \mathbf{0}$ or $E\left\{W'e(\theta)\right\} = \mathbf{0}$
- In $\mathtt{gmm}$ W are exogenous and specified as options
- $e(\theta)$ change at each iteration as θ converges to it's minimum
- The control function approach does not quite fit into $\mathtt{gmm}$'s framework. Let $W(\Pi) = [X, X_2 - Z\Pi]$

$$
\begin{aligned}
E\left\{Z'(X_2 - Z\Pi)\right\} &= \mathbf{0} \\
E\left\{W(\Pi)'\left[y - X_1\beta_1 + X_2\beta + \rho(X_2 - Z\Pi)\right]\right\} &= \mathbf{0}
\end{aligned}
$$

- Estimating $\widehat{\nu}$ and feeding it to $\mathtt{gmm}$ will give incorrect standard errors

GMM for CF

- GMM needs specification $E\{W'e\} = \mathbf{0}$ or $E\{W'e(\theta)\} = \mathbf{0}$
- In gmm W are exogenous and specified as options
- $e(\theta)$ change at each iteration as θ converges to it's minimum
- The control function approach does not quite fit into gmm's framework. Let $W(\Pi) = [X, X_2 - Z\Pi]$

$$\begin{aligned} E\left\{Z'(X_2 - Z\Pi)\right\} &= \mathbf{0} \\ E\left\{W(\Pi)'[y - X_1\beta_1 + X_2\beta + \rho(X_2 - Z\Pi)]\right\} &= \mathbf{0} \end{aligned}$$

- Estimating $\widehat{\nu}$ and feeding it to gmm will give incorrect standard errors

Give gmm what it needs not what it wants

- gmm wants a set of instruments for each equation
- Rewrite the system as

$$
\begin{aligned}
E\left\{Z'(X_2 - Z\Pi)\right\} &= \mathbf{0} \\
E\left\{X'\left[y - X_1\beta_1 + X_2\beta_2 + \rho(X_2 - Z\Pi)\right]\right\} &= \mathbf{0} \\
E\left\{(X_2 - Z\Pi)'\left[y - X_1\beta_1 + X_2\beta_2 + \rho(X_2 - Z\Pi)\right]\right\} &= \mathbf{0}
\end{aligned}
$$

- The last equation satisfies the framework

$$
\begin{aligned}
E\left\{(X_2 - Z\Pi)'\left[y - X_1\beta_1 + X_2\beta_2 + \rho(X_2 - Z\Pi)\right]\right\} &= E\left\{\eta\left(\Pi, \rho, \beta\right)\right\} \\
E\left\{\eta\left(\Pi, \rho, \beta\right)\right\} &= \mathbf{0}
\end{aligned}
$$

- It divided the set of instruments for one of the equations

gmm: Substitutable expressions I

$$\texttt{mpg} = \beta_0 + \beta_1 \texttt{turn} + \beta_2 \texttt{foreign} + \varepsilon$$
$$\texttt{turn} = \pi_0 + \pi_1 \texttt{foreign} + \pi_2 \texttt{weight} + \nu$$

```
. sysuse auto, clear
(1978 Automobile Data)
.
. // Writing down substitutable expression
.
. local zp  {p1}*1.foreign + {p2}*weight + {p0}
. local u   turn -(`zp´)
. local xb  {b1}*turn + {b2}*1.foreign + {b3}*(`u´) + {b0}
. local e   mpg - (`xb´)
.
. // Computing
.
. gmm (eq3: `u´)                                         ///
>     (eq2: (`e´)*(`u´))                                 ///
>     (eq1: `e´),                                        ///
>     instruments(eq3: weight i.foreign)                 ///
>     instruments(eq1: turn i.foreign)                   ///
>     winitial(unadjusted, independent)                  ///
>     from(C) onestep quickderivatives
```

gmm: Substitutable expressions I

$$\begin{aligned}\texttt{mpg} &= \beta_0 + \beta_1 \texttt{turn} + \beta_2 \texttt{foreign} + \varepsilon \\ \texttt{turn} &= \pi_0 + \pi_1 \texttt{foreign} + \pi_2 \texttt{weight} + \nu\end{aligned}$$

```
. sysuse auto, clear
(1978 Automobile Data)

.
. // Writing down substitutable expression
.
. local zp  {p1}*1.foreign + {p2}*weight + {p0}
. local u   turn -(`zp')
. local xb  {b1}*turn + {b2}*1.foreign + {b3}*(`u') + {b0}
. local e   mpg - (`xb')
.
. // Computing
.
. gmm (eq3: `u')                                          ///
>     (eq2: (`e')*(`u'))                                  ///
>     (eq1: `e'),                                         ///
>     instruments(eq3: weight i.foreign)                  ///
>     instruments(eq1: turn i.foreign)                    ///
>     winitial(unadjusted, independent)                   ///
>     from(C) onestep quickderivatives
```

Considerations

- Do not use `gmm` as a computation engine
- You know the fitted values of θ from `regress`
- Use `gmm` to compute standard errors
- Start optimization at `regress` values
- Use `quickderivatives` uses numerical recipes does not go through `deriv()`
- `GMM` is exactly identified, use `onestep`
- Sanity check: `ivregress gmm` should give you the same standard errors and point estimates

gmm: Substitutable expressions II

```
Step 1
Iteration 0:   GMM criterion Q(b) =  1.364e-28
Iteration 1:   GMM criterion Q(b) =  3.309e-29
note: model is exactly identified
GMM estimation
Number of parameters =   7
Number of moments    =   7
Initial weight matrix: Unadjusted                 Number of obs   =         74
```

	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
/p1	-1.809802	.6316408	-2.87	0.004	-3.047795	-.5718085
/p2	.0042183	.0003777	11.17	0.000	.003478	.0049587
/p0	27.44963	1.347933	20.36	0.000	24.80773	30.09153
/b1	-1.56173	.2071108	-7.54	0.000	-1.96766	-1.1558
/b2	-4.476451	1.790227	-2.50	0.012	-7.985232	-.9676693
/b3	1.326564	.2608633	5.09	0.000	.8152814	1.837847
/b0	84.54861	8.641913	9.78	0.000	67.61077	101.4865

```
Instruments for equation eq3: weight 0b.foreign 1.foreign _cons
Instruments for equation eq2: _cons
Instruments for equation eq1: turn 0b.foreign 1.foreign _cons
```

gmm: Starting values

```
. // Getting starting values
. quietly regress turn i.foreign weight
. predict double uhat, residuals
. matrix B = e(b)
. matrix B = B[1,2..colsof(B)]
. quietly regress mpg turn i.foreign uhat
. matrix A = e(b)
. matrix A = A[1,1], A[1,3..colsof(A)]
. matrix C = B,A
```

Evaluator

```
gmm eval [if][in][weight], equations(eqnames)
                            parameters(parameter_names)
                            [youropts stataopts]
```

- I would write an evaluator instead of using substitutable expressions
- Evaluators allow me to add options
- Evaluators are .ado files so they can be used more widely

Evaluator

```
gmm eval [if][in][weight], equations(eqnames)
                           parameters(parameter_names)
                           [youropts stataopts]
```

- I would write an evaluator instead of using substitutable expressions
- Evaluators allow me to add options
- Evaluators are .ado files so they can be used more widely

Example

```
program _cf_linear
        version 16
        syntax varlist if [fweight iweight pweight],           ///
                       at(name)                                ///
                       [                                       ///
                       uhat(varlist)                           ///
                       y1(varname)                             ///
                       y2(varlist)                             ///
                       *                                       ///
                       ]

        tempvar zp xb xbu

        tokenize `varlist'
        local main    `1'
        local reduced `2'
        local aux     `3'

        matrix score double `xb'    = `at' `if', eq(#1)
        matrix score double `zp'    = `at' `if', eq(#2)
        replace `reduced'           = `y2' - `zp' `if'
        replace `uhat'              = `y2' - `zp' `if'    // i.v random
        matrix score double `xbu'   = `at' `if', eq(#3)
        replace `main'              = `y1' - `xb' - `xbu' `if'
        replace `aux'               = (`y1' - `xb'-`xbu')*`uhat' `if'
end
```

Evaluator estimates

```
. gmm _cf_linear, equations(mpg turn uhat)                    ///
>     parameters( "`y1parm´ `y2parm´ uhat:uhat")              ///
>     y1(mpg) y2(turn) uhat(uhat)                              ///
>     instruments(turn: i.foreign weight)                      ///
>     instruments(mpg: i.foreign turn)                         ///
>     winitial(unadjusted, independent)                        ///
>     quickderivatives onestep from(CNEW)
```

Evaluator estimates

```
Iteration 0:   GMM criterion Q(b) =  1.186e-28
Iteration 1:   GMM criterion Q(b) =  6.355e-29
note: model is exactly identified
GMM estimation
Number of parameters =   7
Number of moments    =   7
Initial weight matrix: Unadjusted                 Number of obs   =         74
------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
mpg          |
        turn |   -1.56173   .2071109    -7.54   0.000     -1.96766     -1.1558
             |
     foreign |
    Foreign  |  -4.476451   1.790228    -2.50   0.012    -7.985233   -.9676679
       _cons |   84.54861   8.641919     9.78   0.000     67.61076    101.4865
-------------+----------------------------------------------------------------
turn         |
     foreign |
    Foreign  |  -1.809802   .6316412    -2.87   0.004    -3.047796   -.5718077
      weight |   .0042183   .0003777    11.17   0.000      .003478    .0049587
       _cons |   27.44963   1.347934    20.36   0.000     24.80773    30.09154
-------------+----------------------------------------------------------------
uhat         |
        uhat |   1.326564   .2608634     5.09   0.000     .8152812    1.837847
------------------------------------------------------------------------------
Instruments for equation mpg: 0b.foreign 1.foreign turn _cons
Instruments for equation turn: 0b.foreign 1.foreign weight _cons
Instruments for equation uhat: _cons
```

Sanity check

```
. estimates store gmm
. quietly ivregress gmm mpg i.foreign (turn = weight)
. estimates store ivreg_gmm
. estimates table gmm ivreg_gmm, eq(1) keep(turn 1.foreign _cons) se
```

Variable	gmm	ivreg_gmm
turn	-1.5617299	-1.5617299
	.2071109	.2071109
foreign		
Foreign	-4.4764507	-4.4764507
	1.7902282	1.7902282
_cons	84.548613	84.548613
	8.641919	8.641919

legend: b/se

A command

```
cfunction estimator y Xs ..., endogenous(end1 ...  endk = ...)
                              cfvar([newvars], [...])  ...
```

```
cfunction estimator y Xs ..., endogenous(end1 = ...)  ...
                              endogenous(endk = ...)
                              cfvar([newvars],[...])  ...
```

- *estimator* is `probit`, `linear`, ...
- `endogenous()` might be multiple equations with different instruments
- A variable is created with the residuals of the first step
- `cfvar(`*newvars*`, [replace float])`

A command

```
cfunction estimator y Xs ..., endogenous(end1 ...  endk = ...)
                              cfvar([newvars], [...])  ...
```

```
cfunction estimator y Xs ..., endogenous(end1 = ...)  ...
                              endogenous(endk = ...)
                              cfvar([newvars],[...])  ...
```

- *estimator* is `probit`, `linear`, ...
- `endogenous()` might be multiple equations with different instruments
- A variable is created with the residuals of the first step
- `cfvar(`*newvars*`, [replace float])`

The command

```
. cfunction linear mpg i.foreign, endogenous(turn = i.foreign weight)
Iteration 0:   EE criterion =  7.381e-29
Iteration 1:   EE criterion =  3.125e-29
Control-function linear regression              Number of obs     =         74
Outcome model          : regress
Control-function model: regress
------------------------------------------------------------------------------
             |               Robust
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
mpg          |
        turn |   -1.56173   .2071356    -7.54   0.000    -1.967708   -1.155752
             |
     foreign |
    Foreign  |  -4.476451   1.790523    -2.50   0.012    -7.985812   -.9670892
             |
       _cons |   84.54861    8.64284     9.78   0.000     67.60896    101.4883
------------------------------------------------------------------------------
```

The command

```
. cfunction linear mpg i.foreign, endogenous(turn=i.foreign weight) aequations
Iteration 0:   EE criterion =  7.381e-29
Iteration 1:   EE criterion =  3.125e-29
Control-function linear regression              Number of obs     =         74
Outcome model         : regress
Control-function model: regress

------------------------------------------------------------------------------
             |               Robust
         mpg |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
mpg          |
        turn |   -1.56173   .2071356    -7.54   0.000    -1.967708   -1.155752
             |
     foreign |
    Foreign  |  -4.476451   1.790523    -2.50   0.012    -7.985812   -.9670892
             |
       _cons |   84.54861    8.64284     9.78   0.000     67.60896    101.4883
-------------+----------------------------------------------------------------
turn         |
     foreign |
    Foreign  |  -1.809802   .6316412    -2.87   0.004    -3.047796   -.5718077
             |
      weight |   .0042183   .0003777    11.17   0.000      .003478    .0049587
       _cons |   27.44963   1.347934    20.36   0.000     24.80773    30.09154
-------------+----------------------------------------------------------------
__Cf_V__a_R1 |
       _cons |   1.326564   .2609407     5.08   0.000     .8151298    1.837998
------------------------------------------------------------------------------
```

Model II: Fractional (binary) outcomes

$$
\begin{aligned}
E(y|X,\nu) &= \Phi(X_1\beta_1 + X_2\beta_2 + \rho\nu) \\
X_2 &= Z\Pi + \nu
\end{aligned}
$$

- You can think of y as $y = X_1\beta_1 + X_2\beta_2 + \varepsilon > 0$ and (ε, ν) being correlated and jointly normal
- y could be described by another model as long as X_2 is continuous and endogeneity is due to a relation of ε and ν
- Z are unrelated to ν

Interpretation

- Coefficients and standard errors are asymptotically equivalent to two-step estimates computed by `ivprobit ..., twostep`
- Coefficients should not be taken too seriously. What is important is to think about effects. (Editorial comment).
- We will be able to compute effects because the coefficient vectors and standard errors are kept for all equations

The command

```
. cfunction probit foreign mpg, endogenous(headroom = mpg weight)
Iteration 0:   EE criterion =  1.525e-24
Iteration 1:   EE criterion =  1.065e-31
Control-function fractional regression          Number of obs     =         74
Outcome model          : fracreg probit
Control-function model: regress
```

foreign	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
foreign						
headroom	-5.017474	2.1125	-2.38	0.018	-9.157898	-.877051
mpg	-.1544128	.1484151	-1.04	0.298	-.445301	.1364754
_cons	17.31753	9.024816	1.92	0.055	-.370786	35.00584

Sanity check I

```
. quietly cfunction probit foreign mpg, endogenous(headroom = mpg weight)
. estimates store cfprobit
. quietly ivprobit foreign mpg (headroom = weight), twostep
. estimates store ivtwo
. estimates table cfprobit ivtwo, se eq(1) drop(`drop´)
```

Variable	cfprobit	ivtwo
headroom	-5.0174745	-5.0174745
	2.1124998	2.4034212
mpg	-.1544128	-.1544128
	.14841505	.15947795
_cons	17.317527	17.317528
	9.0248155	10.182072

legend: b/se

Sanity check II

```
. webuse cattaneo2, clear
(Excerpt from Cattaneo (2010) Journal of Econometrics 155: 138-154)
. quietly cfunction probit lbweight i.msmoke i.alcohol mage,           ///
>     endogenous(medu = i.foreign fedu i.msmoke i.alcohol mage)
. estimates store uno
. quietly ivprobit lbweight i.msmoke i.alcohol mage (medu = i.foreign fedu)
. estimates store dos
. estimates table uno dos, keep(lbweight:) se

------------------------------------------
    Variable |     uno           dos
-------------+----------------------------
        medu | -.02132736    -.02130645
             |  .02605357     .02640689
             |
      msmoke |
  1-5 daily  |  .01475317     .01475978
             |  .15390946     .14990178
 6-10 daily  |  .51695718     .51685433
             |  .09787212     .09627577
  11+ daily  |  .39551778     .39544583
             |  .10403561     .10282816
             |
     alcohol |
          1  |  .18803644     .18800059
             |  .14087207     .14518304
             |
        mage | -.01016345    -.01016394
             |  .00732448     .00697637
       _cons | -1.1213499    -1.121227
             |  .26433385     .25888826
------------------------------------------
                              legend: b/se
```

General postestimation considerations

- Construct predictions as a function of `cfvar`
- Return `e(covariates)`. Variables `margins` operates over.
  - Exclude `cfvar`
  - Exclude excluded instruments (z's)
- `margins` only perturbs variables in `e(covariates)`
- The coefficients on `cfvar` provide test for endogeneity
- `margins` quantities have a structural function interpretation

Model III: Panel exponential mean

$$
\begin{aligned}
E\left(y_{it}|X_{it1}, X_{it2}, \alpha_i, \nu_{it}\right) &= \exp(X_{it1}\beta_1 + X_{it2}\beta_2 + \alpha_i + \rho\nu_{it}) \\
X_{it2} &= Z_{it}\Pi + \gamma_i + \nu_{it}
\end{aligned}
$$

- Fit a fixed effects regression of X_{it2} on Z_{it} or a correlated random effects estimator (i.e. Mundlak) and get residuals
- Compute fixed effects Poisson regression including residuals
- Other estimators and conditions can be considered. They will imply different GMM estimators (i.e Windmeijer (2000) and Lin and Wooldridge (2019))

Moment conditions

- $\widetilde{W} \equiv W_{it} - \overline{W}_i + \overline{W}$
- $\left(\widetilde{X}_2 - \widetilde{Z}\Pi\right) \equiv \nu(\Pi)$
- $\exp(X_1\beta_1 + X_2\beta_2 + \rho\nu(\Pi)) \equiv \theta$

$$
\begin{aligned}
E\left\{\widetilde{Z}'\nu(\Pi)\right\} &= \mathbf{0} \\
E\left\{X'\left[y - \frac{\overline{y}}{\overline{\overline{\theta}}}\theta\right]\right\} &= \mathbf{0} \\
E\left\{\nu(\Pi)'\left[y - \frac{\overline{y}}{\overline{\overline{\theta}}}\theta\right]\right\} &= \mathbf{0}
\end{aligned}
$$

Evaluator

```
program cfxtpoisson
        version 16
        syntax varlist if [fweight iweight pweight],            ///
                        at(name)                                 ///
                        [                                        ///
                        at(name)                                 ///
                        id(string)                               ///
                        uhat(varlist)                            ///
                        y1(varname)                              ///
                        y2(varlist)                              ///
                        *                                        ///
                        ]
        tempvar zp xb xbu xbbar ybar
        tokenize `varlist'
        local main    `1'
        local reduced `2'
        local aux     `3'

        matrix score double `xb'    = `at' `if', eq(#1)
        matrix score double `zp'    = `at' `if', eq(#2)
        replace `reduced'           = `y2' - `zp' `if'
        replace `uhat'              = `y2' - `zp' `if'     // i.v random
        matrix score double `xbu'   = `at' `if', eq(#3)

        replace `xb'                = exp(`xb')
        replace `xbu'               = exp(`xbu')
        egen double `xbbar'         = mean(`xb'*`xbu') `if', by(`id')
        egen double `ybar'          = mean(`y1') `if', by(`id')
        replace `main'              = `y1' -`xb'*`xbu'*`ybar'/`xbbar' `if'
        replace `aux'               = (`main')*`uhat' `if'
end
```

Exponential mean simulated data

$$\begin{aligned} \texttt{y2} &= 1 - \texttt{x1} + \texttt{x2 - z1 + z2 - z3 + u1 + a} \\ \texttt{y1} &= \texttt{exp(.5(1 - y2 + x1 - x2) + u2 + a)} \end{aligned}$$

- u1 and u2 are correlated jointly normal time-varying unobservables
  - Correlation $\rho = .7$
- a is a time invariant unobservable and normal correlated with covariates
- All covariates are standardized chi-squares with 5 degrees of freedom

iterlogonly

```
. gmm cfxtpoisson, equations(y1 y2 uhat) id(id)                    ///
>                  parameters("`y1parm´ `y2parm´ uhat:uhat")        ///
>                  y2(dmy2) y1(y1) uhat(uhat)                       ///
>                  instruments(y2: dmz1 dmz2 dmz3 dmx1 dmx2)        ///
>                  instruments(y1: y2 x1 x2, nocons)                ///
>                  winitial(unadjusted, independent)                ///
>                  quickderivatives onestep from(C) iterlogonly     ///
>                  vce(cluster id)
Iteration 0:   GMM criterion Q(b) =  2.696e-20
Iteration 1:   GMM criterion Q(b) =  9.749e-31
```

Results

```
. gmm
GMM estimation
Number of parameters =   10
Number of moments    =   10
Initial weight matrix: Unadjusted                 Number of obs   =     10,000
                                 (Std. Err. adjusted for 2,000 clusters in id)
------------------------------------------------------------------------------
             |               Robust
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y1           |
          y2 |  -.5150497   .0183471   -28.07   0.000    -.5510094     -.47909
          x1 |   .4935781   .0308917    15.98   0.000     .4330315    .5541247
          x2 |  -.4997544   .0388162   -12.87   0.000    -.5758327    -.423676
-------------+----------------------------------------------------------------
y2           |
        dmz1 |  -.9914454   .0111035   -89.29   0.000    -1.013208    -.969683
        dmz2 |   .9952376   .0115265    86.34   0.000     .9726461    1.017829
        dmz3 |  -1.000285   .0116556   -85.82   0.000    -1.023129   -.9774401
        dmx1 |    -1.0056   .0110415   -91.07   0.000    -1.027241    -.983959
        dmx2 |   .9985457   .0109718    91.01   0.000     .9770413     1.02005
       _cons |   .9681836    .000386  2508.14   0.000      .967427    .9689402
-------------+----------------------------------------------------------------
uhat         |
        uhat |   .7922526   .0372532    21.27   0.000     .7192377    .8652675
------------------------------------------------------------------------------
Instruments for equation y1: y2 x1 x2
Instruments for equation y2: dmz1 dmz2 dmz3 dmx1 dmx2 _cons
Instruments for equation uhat: _cons
```

A preliminary simulation exercise

Estimator	Bias	Coverage rate
`cfunction_linear`	-0.000	0.950
`cfunction_probit`	-0.001	0.953
`xtcfunction`	0.000	0.945

- $\rho = .5$
- Endogeneity comes from:
  - Joint normality of time invariant unobservables (all)
  - Common component in endogenous covariate and time-invariant unobservables
- All covariates are standardized chi-square with 5 degrees of freedom
- N = 3000, T = 5 for cross section I kept one time period (T = 2)
- Results are from 1000 draws
- No time to compare to bootstrap

Conclusion

- I illustrated how to use `gmm` to compute control function estimates and their standard errors
- Along the way I illustrated some tools for those wanting to use `gmm` more efficiently
- Control function GMM standard error estimates are an attractive alternative to bootstrap standard errors
- The estimators open up the possibilities of using `margins` for control function estimates described