Dealing with Heterogeneity: Finite Mixture Models
處理群體異質性: 有限混入模型

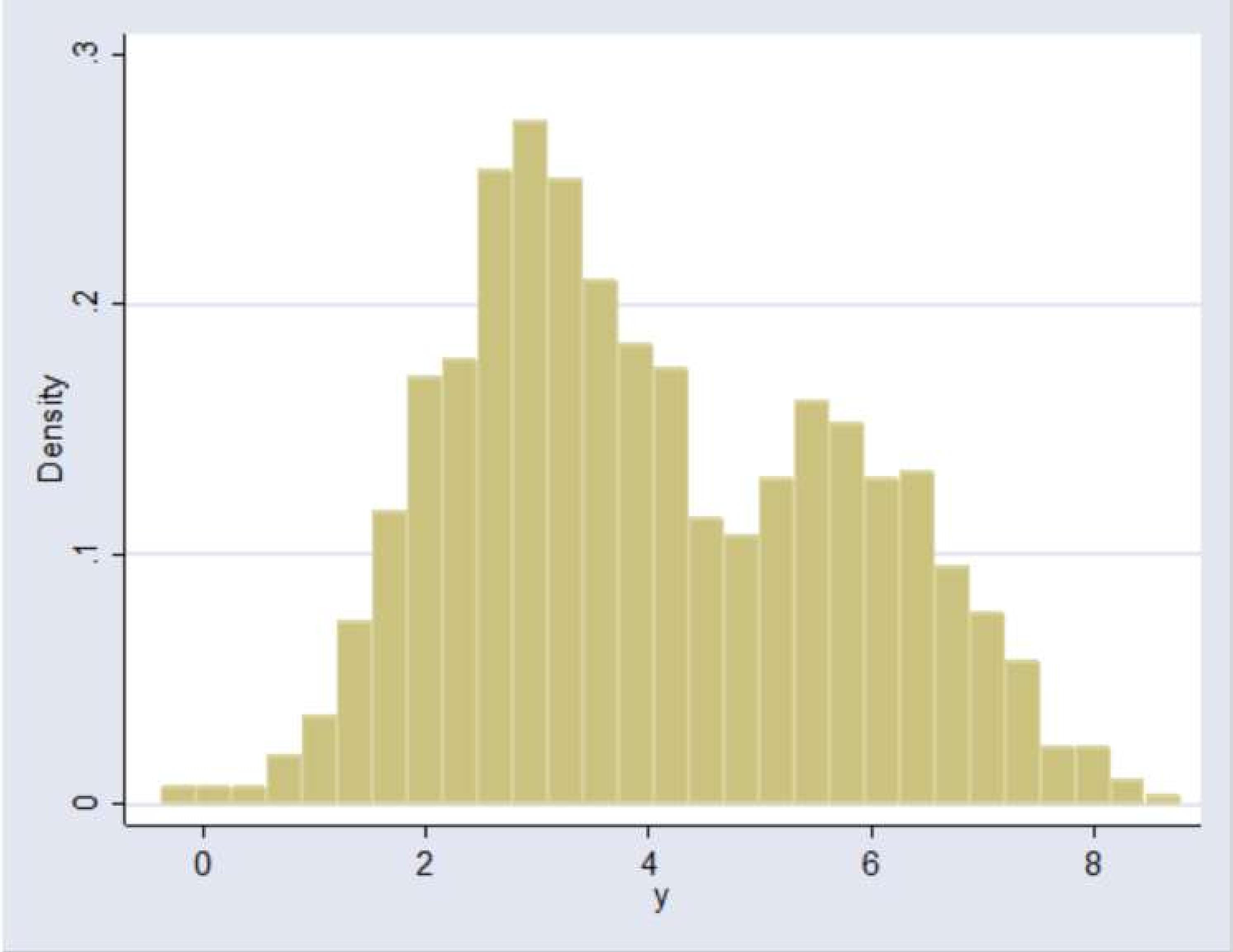
Joseph Tao-yi Wang (王道一)
EEBGT, Experimetrics Module 6

Part I: Mixture of Two Normal Distributions
第一部分: 混入兩個常態分配

Joseph Tao-yi Wang (王道一)
EEBGT, Experimetrics Module 6

Mixture of Two Normal Distributions

- Data (N=1,000)
  `mixture_sim.dta`
- STATA Command:
  `hist y`
- STATA Results:
  - 2 Types of Subjects?
  - Mean at 3 and 6?



Mixture of Two Normal Distributions

- Type 1: Mixing Proportion $\Pr(\text{Type 1}) = p$
  - Choose $y \sim N(\mu_1, \sigma_1^2)$ with $f(y|\text{Type 1}) = \frac{1}{\sigma_1}\phi\left(\frac{y-\mu_1}{\sigma_1}\right)$
- Type 2: Mixing Proportion $\Pr(\text{Type 2}) = (1-p)$
  - Choose $y \sim N(\mu_2, \sigma_2^2)$ with $f(y|\text{Type 2}) = \frac{1}{\sigma_2}\phi\left(\frac{y-\mu_2}{\sigma_2}\right)$
- Marginal Density (Likelihood):

$$f(y;\underline{\mu_1,\sigma_1,\mu_2,\sigma_2,p}) = p\cdot\frac{1}{\sigma_1}\phi\left(\frac{y-\mu_1}{\sigma_1}\right) + (1-p)\cdot\frac{1}{\sigma_2}\phi\left(\frac{y-\mu_2}{\sigma_2}\right)$$

Mixture of Two Normal Distributions

- Estimate $\hat{\mu_1}, \hat{\sigma_1}, \hat{\mu_2}, \hat{\sigma_2}, \hat{p}$ to max.
- Sample log-Likelihood: $\log L = \sum_{i=1}^{n} \ln f(y_i; \mu_1, \sigma_1, \mu_2, \sigma_2, p)$
  - (for y_1, y_2, … , y_n)
- Calculate Posterior Probability:

$$\Pr(\text{Type } 1|y) = \frac{f(y|\text{Type } 1)\Pr(\text{Type } 1)}{f(y|\text{Type } 1)\Pr(\text{Type } 1) + f(y|\text{Type } 2)\Pr(\text{Type } 2)}$$

$$= \frac{p \cdot \frac{1}{\sigma_1}\phi\left(\frac{y-\mu_1}{\sigma_1}\right)}{p \cdot \frac{1}{\sigma_1}\phi\left(\frac{y-\mu_1}{\sigma_1}\right) + (1-p) \cdot \frac{1}{\sigma_2}\phi\left(\frac{y-\mu_2}{\sigma_2}\right)}$$

STATA Code: Components of Log-Likelihood

- `mu1,mu2,sig1,sig2,p`: $\hat{\mu_1}, \hat{\sigma_1}, \hat{\mu_2}, \hat{\sigma_2}, \hat{p}$
  - `f1`: $f(y|\text{Type 1}) = \frac{1}{\sigma_1}\phi\left(\frac{y-\mu_1}{\sigma_1}\right)$
  - `f2`: $f(y|\text{Type 2}) = \frac{1}{\sigma_2}\phi\left(\frac{y-\mu_2}{\sigma_2}\right)$
- `logl`:

$$\ln[f(y)] = \ln\left[p \cdot \frac{1}{\sigma_1}\phi\left(\frac{y-\mu_1}{\sigma_1}\right) + (1-p) \cdot \frac{1}{\sigma_2}\phi\left(\frac{y-\mu_2}{\sigma_2}\right)\right]$$

  - `postp1`: $\Pr(\text{Type 1})$
  - `postp2`: $\Pr(\text{Type 2})$

STATA Code: Components of Log-Likelihood

```
program drop _all
* LIKELIHOOD EVALUATION PROGRAM STARTS HERE:
program define mixture
args logl mu1 sig1 mu2 sig2 p
tempvar f1 f2

* GENERATE TYPE-CONDITIONAL DENSITIES:
quietly gen double `f1'=(1/`sig1')*normalden((y-`mu1')/`sig1')
quietly gen double `f2'=(1/`sig2')*normalden((y-`mu2')/`sig2')

* COMBINE TYPE-CONDITIONAL DENSITIES WITH MIXING PROPORTIONS TO GENERATE MARGINAL DENSITY.
* THIS IS THE FUNCTION THAT NEEDS TO BE MAXIMISED WHEN SUMMED OVER THE SAMPLE:
quietly replace `logl'=ln(`p'*`f1'+(1-`p')*`f2')

* GENERATE THE POSTERIOR TYPE PROBABILITIES, AND MAKE THEM AVAILABLE OUTSIDE THE PROGRAM:
quietly replace postp1=`p'*`f1'/(`p'*`f1'+(1-`p')*`f2')
quietly replace postp2=(1-`p')*`f2'/(`p'*`f1'+(1-`p')*`f2')
quietly putmata postp1, replace
```

Global Variable: y
Local Variable: `mu1',`sig1',...

```
program drop _all
* LIKELIHOOD EVALUATION PROGRAM STARTS HERE:
program define mixture
args logl mu1 sig1 mu2 sig2 p
tempvar f1 f2

* GENERATE TYPE-CONDITIONAL DENSITIES:
quietly gen double ‘f1’=(1/‘sig1’)*normalden((y-‘mu1’)/‘sig1’)
quietly gen double ‘f2’=(1/‘sig2’)*normalden((y-‘mu2’)/‘sig2’)

* COMBINE TYPE-CONDITIONAL DENSITIES WITH MIXING PROPORTIONS TO GENERATE MARGINAL DENSITY.
* THIS IS THE FUNCTION THAT NEEDS TO BE MAXIMISED WHEN SUMMED OVER THE SAMPLE:
quietly replace ‘logl’=ln(‘p’*‘f1’+(1-‘p’)*‘f2’)

* GENERATE THE POSTERIOR TYPE PROBABILITIES, AND MAKE THEM AVAILABLE OUTSIDE THE PROGRAM:
quietly replace postp1=‘p’*‘f1’/(‘p’*‘f1’+(1-‘p’)*‘f2’)
quietly replace postp2=(1-‘p’)*‘f2’/(‘p’*‘f1’+(1-‘p’)*‘f2’)
quietly putmata postp1, replace
quietly putmata postp2, replace
end
* END OF LIKELIHOOD EVALUATION PROGRAM
* READ DATA:
use mixture_sim, clear
```

Save postp1, postp2 with STATA mata command putmata for later use

STATA: Mixture of Two Normal Distributions

- STATA Code:

```
* INITIALISE TWO POSTERIOR PROBABILITY VARIABLES:
gen postp1=.
gen postp2=.

* SPECIFY STARTING VALUES, AND APPLY ML:
mat start=(3, 1.5, 6, 1.5, .5)
ml model lf mixture /mu1 /sig1 /mu2 /sig2 /p
ml init start, copy
ml maximize

* EXTRACT POSTERIOR TYPE PROBABILITY, AND PLOT THEM
drop postp1 postp2
```

Assign Initial Values by Plotting `hist y`, or Using Results From Linear Regressions

STATA: Mixture of Two Normal Distributions

- STATA Results:

Wald chi2(0) = .
Log likelihood = -1908.2805 Prob > chi2 = .

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
eq1	$\hat{\mu}_1 = 2.982\ (0.074)$					
_cons	2.981757	.0743116	40.13	0.000	2.836109	3.127405
eq2	$\hat{\sigma}_1 = 1.015\ (0.050)$					
_cons	1.014725	.0499721	20.31	0.000	.9167818	1.112669
eq3	$\hat{\mu}_2 = 5.950\ (0.116)$					
_cons	5.950353	.1158028	51.38	0.000	5.723384	6.177322
eq4	$\hat{\sigma}_2 = 0.977\ (0.072)$					
_cons	.9768525	.0721166	13.55	0.000	.8355064	1.118198
eq5	$\hat{p} = 0.649\ (0.030)$					
_cons	.6494311	.0296983	21.87	0.000	.5912235	.7076387

Population has 64.9% from $N(2.98,\ 1.02^2)$ 35.1% from $N(5.95,\ 0.98^2)$

STATA: Mixture of Two Normal Distributions

- STATA Code:

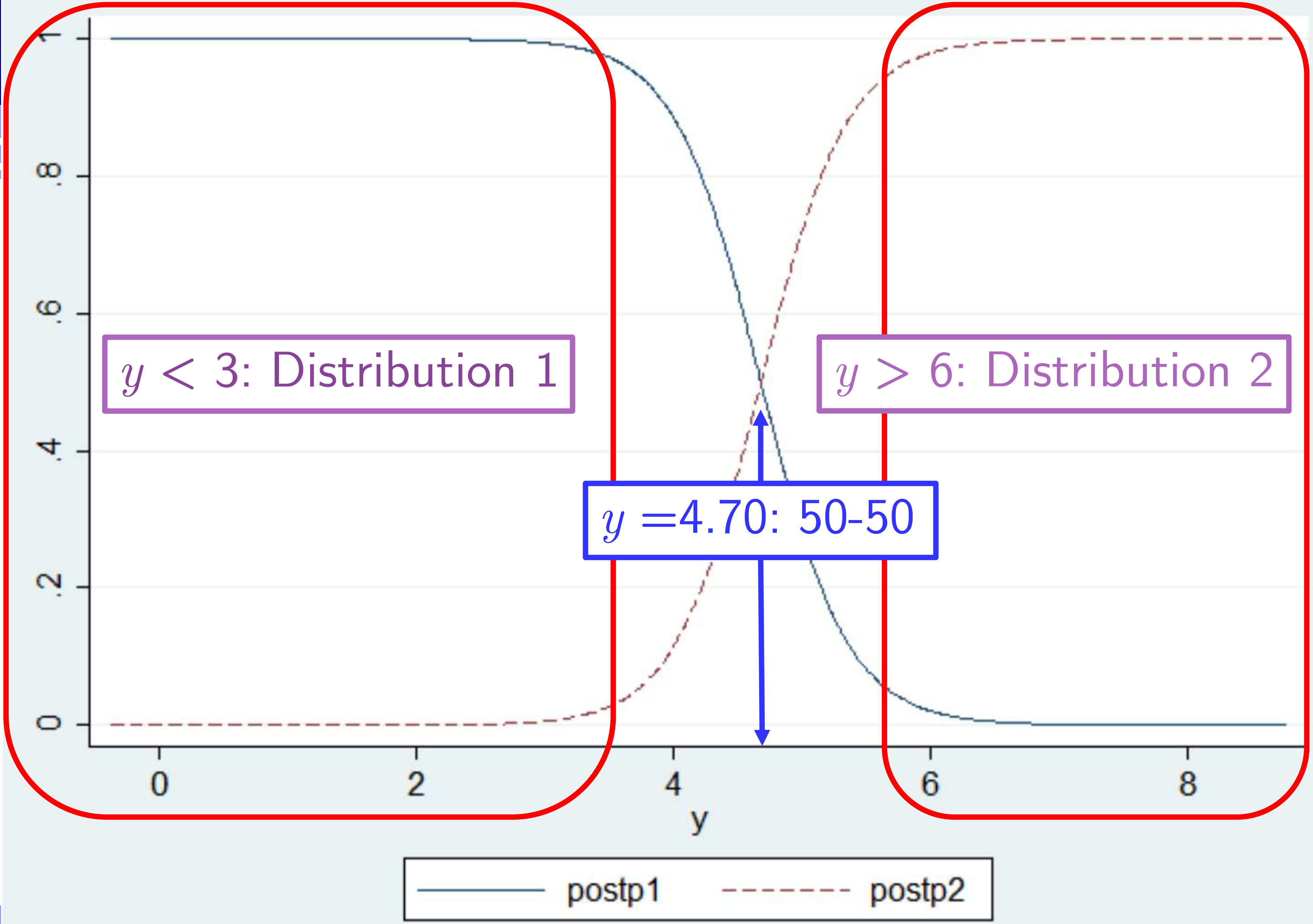
```
* EXTRACT POSTERIOR TYPE PROBABILITY, AND PLOT THEM AGAINST y:
drop postp1 postp2
getmata postp1
getmata postp2
sort y
line postp1 postp2 y , lpattern(l -)
```

Retrieve Temporary Variables `postp1`, `postp2` with STATA mata command `getmata`

Plot Posterior Probability vs. y

STATA: Mix

- STATA
  Results:



Finite Mixture Model STATA Command: fmm

- STATA `fmm 2: regress y`

  Results: 2 Types | Model of Each Type: Regress on Intercept

```
Fitting class model:
Iteration 0:   (class) log likelihood = -693.14718
Iteration 1:   (class) log likelihood = -693.14718

Fitting outcome model:
Iteration 0:   (outcome) log likelihood = -1340.7846
Iteration 1:   (outcome) log likelihood = -1340.7846

Refining starting values:
```

Finite Mixtu

- STATA
  Results:

```
Refining starting values:
Iteration 0:   (EM) log likelihood = -2114.989
Iteration 1:   (EM) log likelihood = -2144.1684
Iteration 2:   (EM) log likelihood = -2155.951
Iteration 3:   (EM) log likelihood = -2159.9264
Iteration 4:   (EM) log likelihood = -2159.9464
Iteration 5:   (EM) log likelihood = -2157.8613
Iteration 6:   (EM) log likelihood = -2154.6472
Iteration 7:   (EM) log likelihood = -2150.8481
Iteration 8:   (EM) log likelihood = -2146.7758
Iteration 9:   (EM) log likelihood = -2142.6116
Iteration 10:  (EM) log likelihood = -2138.4622
Iteration 11:  (EM) log likelihood = -2134.3904
Iteration 12:  (EM) log likelihood = -2130.4335
Iteration 13:  (EM) log likelihood = -2126.6137
```

Finite Mixtu

- STATA
  Results:

```
Iteration 14:   (EM) log likelihood = -2122.9441
Iteration 15:   (EM) log likelihood = -2119.432
Iteration 16:   (EM) log likelihood = -2116.0816
Iteration 17:   (EM) log likelihood = -2112.8942
Iteration 18:   (EM) log likelihood = -2109.8699
Iteration 19:   (EM) log likelihood = -2107.0071
Iteration 20:   (EM) log likelihood = -2104.3034
Note: EM algorithm reached maximum iterations.

Fitting full model:
Iteration 0:   log likelihood = -1909.8137
Iteration 1:   log likelihood = -1908.4031
Iteration 2:   log likelihood = -1908.2811
Iteration 3:   log likelihood = -1908.2805
Iteration 4:   log likelihood = -1908.2805
```

Finite Mixtu

- STATA Results:

```
Finite mixture model                            Number of obs     =      1,000
Log likelihood = -1908.2805

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
1.Class      |  (base outcome)
-------------+----------------------------------------------------------------
2.Class      |
       _cons |  -.6165402    .130444    -4.73   0.000    -.8722058   -.3608746
------------------------------------------------------------------------------

Class          : 1
Response       : y
Model          : regress

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
       _cons |   2.981758   .0743115    40.13   0.000      2.83611    3.127406
-------------+----------------------------------------------------------------
    var(e.y)|   1.029668   .1014158                       .848905    1.248921
------------------------------------------------------------------------------

Class          : 2
Response       : y
```

$\hat{p} = 0.617\ (0.130)$

$\hat{\mu}_1 = 2.982\ (0.074)$

$\hat{\sigma}_1 = 1.030\ (0.101)$

Finite Mixtu

- STATA Results:

Results Similar to MLE estimation!!

predict yields the same posterior type probabilities!!

```
Class          : 1
Response       : y
Model          : regress

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
       _cons |   2.981758   .0743115    40.13   0.000      2.83611    3.127406
-------------+----------------------------------------------------------------
    var(e.y)|   1.029668   .1014158                       .848905    1.248921
------------------------------------------------------------------------------

Class          : 2
Response       : y
Model          : regress

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
       _cons |   5.950353   .1158024    51.38   0.000     5.723385    6.177322
-------------+----------------------------------------------------------------
    var(e.y)|   .9542398   .1408942                      .7144585    1.274495
------------------------------------------------------------------------------
```

$\hat{\mu}_1 = 2.982\ (0.074)$

$\hat{\sigma}_1 = 1.030\ (0.101)$

```
predict post1 , pos eq(component1)
predict post2 , pos eq(component2)
```

$\hat{\mu}_2 = 5.950\ (0.116)$

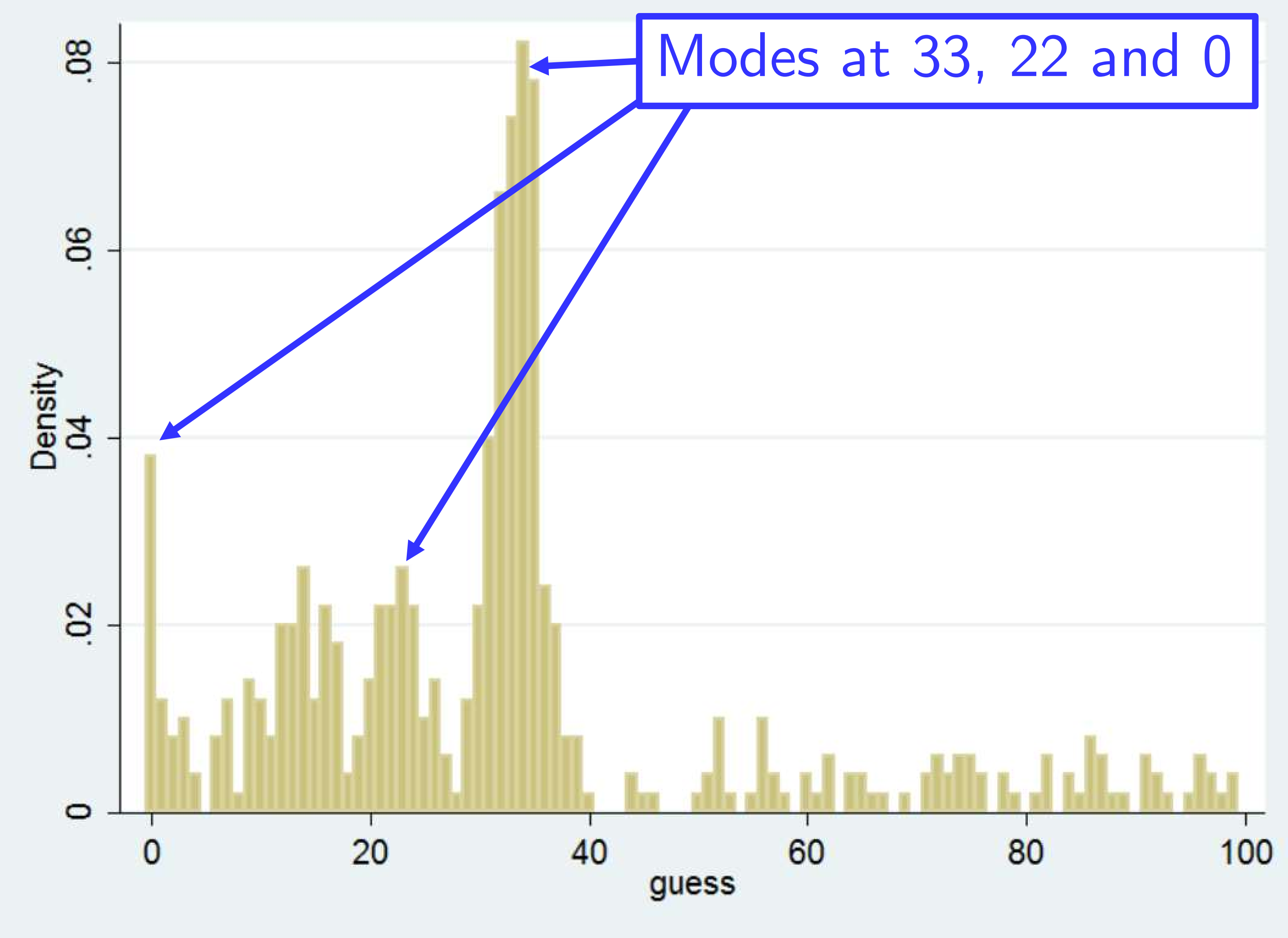
$\hat{\sigma}_2 = 0.954\ (0.141)$

Part II: A Level-k Model For The Beauty Contest Game
第二部分: 選美預測賽局的多層次認知模型

Joseph Tao-yi Wang (王道一)
EEBGT, Experimetrics Module 6

The p-Beauty Contest Game: Nagel (1995)

- Choose a whole number in 0-100
- Number Closest to "p=2/3 of the Average" wins
  - Simulated Data of N=500 Players: `beauty_sim.dta`



A Level-k Model For the Beauty Contest Game

  - Level-0 Reasoners Choose Randomly from $\text{Unif}[0,100]$
- Level-1 Believe Others are Level-0 and Choose 33
  - Mean Guess = 50 and 50 x (2/3) = 33.333
- Level-2 Believe Others are Level-1 and Choose 22
  - Mean Guess = 33 and 33 x (2/3) = 22
- Level-3 Believe Others are Level-2 and Choose 15
  - Mean Guess = 22 and 22 x (2/3) = 14.667
- Level-4 Believe Others are Level-3 and Choose 10, etc.

A Level-k Model For the Beauty Contest Game

- If All Subjects Believe Others are Level-K, $K \to \infty$
  - All Guess 0 and Have Equal Chance to Win
- Same as Nash Equilibrium!
  - But real subjects do NOT play Nash (at least initially)
- To Estimate the Level-k Model:
  - Assume the Maximum Level $= J$
  - Let Level-J = naive-Nash (Choose Nash)
  - Let Level-0 choose randomly from uniform distribution

Estimating the Level-k Model

- Level-j Chooses: $y|_{\text{Type } j} = y_j^* + \epsilon,\ \epsilon \sim N(0, \sigma^2)$
  - Where $y_j^* =$ best guess of Type j $(j = 1,\ldots,J)$
- Conditional Density Functions:
  - Level-0: $f(y|L_0) = 1/100,\ 0 \le y \le 100$
  - Level-j: $f(y|L_j) = \frac{1}{\sigma}\phi\left(\frac{y - y_j^*}{\sigma}\right),\ 0 \le y \le 100$ $(j = 1,\ldots,J)$
- Sample Log-Likelihood:
  - For $y_i,\ i = 1,\ldots,n$: $\log L = \sum_{i=1}^{n} \ln\left[\frac{p_0}{100} + \sum_{j=1}^{J} p_j \frac{1}{\sigma}\phi\left(\frac{y_i - y_j^*}{\sigma}\right)\right]$
  - Mixture $(p_0, p_1, \ldots, p_J)$

Estimating the Level-k

- $J = 5$
  - STATA: Maximized Log-Likelihood
- Best Guesses:
  - ${y_1}^* = 33.5$
  - ${y_2}^* = 22.4$
  - ${y_3}^* = 15.0$
  - ${y_4}^* = 10.1$
  - ${y_5}^* = 0$ (Naïve Nash)

```
program define beauty_mixture
args lnf p1 p2 p3 p4 p5 sig
tempvar f0 f1 f2 f3 f4 f5 l

quietly{

gen double `f0'=0.01
gen double `f1'=(1/`sig')*normalden((y-33.5)/`sig')
gen double `f2'=(1/`sig')*normalden((y-22.4)/`sig')
gen double `f3'=(1/`sig')*normalden((y-15.0)/`sig')
gen double `f4'=(1/`sig')*normalden((y-10.1)/`sig')
gen double `f5'=(1/`sig')*normalden((y-0)/`sig')

gen double `l'=(1-`p1'-`p2'-`p3'-`p4'-`p5')*`f0' ///
+`p1'*`f1'+`p2'*`f2'+`p3'*`f3'+`p4'*`f4'+`p5'*`f5'

replace postp0=(1-`p1'-`p2'-`p3'-`p4'-`p5')*`f0'/`l'
replace postp1=`p1'*`f1'/`l'
replace postp2=`p2'*`f2'/`l'
replace postp3=`p3'*`f3'/`l'
replace postp4=`p4'*`f4'/`l'
replace postp5=`p5'*`f5'/`l'
```

Estimating the Level-k

- $J = 5$
  - STATA: Maximized Log-Likelihood
- Best Guesses:
  - $y_1{}^* = 33.5$
  - $y_2{}^* = 22.4$
  - $y_3{}^* = 15.0$
  - $y_4{}^* = 10.1$
  - $y_5{}^* = 0$ (Naïve Nash)

```
replace `lnf'=ln((1-`p1'-`p2'-`p3'-`p4'-`p5')*`f0' ///
+`p1'*`f1'+`p2'*`f2'+`p3'*`f3'+`p4'*`f4'+`p5'*`f5')

putmata postp0, replace
putmata postp1, replace
putmata postp2, replace
putmata postp3, replace
putmata postp4, replace
putmata postp5, replace

}

end



gen postp0=.
gen postp1=.
gen postp2=.
gen postp3=.
gen postp4=.
gen postp5=.
```

Estimating the Level-k Model

- Estimate $p_1, p_2, p_3, p_4, p_5, \sigma$

```
mat start=(0.3, 0.4, 0.1, 0.1,0.05, 2)

ml model lf beauty_mixture /p1 /p2 /p3 /p4 /p5 /sig
ml init start, copy
ml maximize
nlcom p0: 1-_b[p1:_cons]-_b[p2:_cons]-_b[p3:_cons]-_b[p4:_cons]-_b[p5:_cons]

drop postp*

getmata postp0
getmata postp1
getmata postp2
getmata postp3
getmata postp4
getmata postp5
sort y
line postp0 postp1 postp2 postp3 postp4 postp5 y , lpattern(- l l l l l)
```

Estimating t

- STATA Results:

40% are Level-1, 11% are Level-2, 9% are Level-3 (5% Naïve Nash)

```
                                                  Number of obs   =        500
                                                  Wald chi2(0)    =          .
Log likelihood = -1985.0613                       Prob > chi2     =          .
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
p1 _cons	.3982665	.023804	16.73	0.000	.3516116	.4449213
p2 _cons	.1128533	.0163975	6.88	0.000	.0807148	.1449919
p3 _cons	.0898775	.0159347	5.64	0.000	.0586461	.121109
p4 _cons	.0462681	.0135852	3.41	0.001	.0196415	.0728946
p5 _cons	.0500939	.0117892	4.25	0.000	.0269876	.0732002
sig _cons	1.929627	.1027345	18.78	0.000	1.728271	2.130982

$\hat{p_1} = 0.398\ (0.024)$

$\hat{p_2} = 0.113\ (0.016)$

$\hat{p_3} = 0.090\ (0.016)$

$\hat{p_4} = 0.046\ (0.014)$

$\hat{p_5} = 0.050\ (0.012)$

$\hat{\sigma} = 1.930\ (0.103)$

Estimating the Level-k Model

- Estimate
  $p_1, p_2, p_3, p_4, p_5, \sigma$

```
mat start=(0.3, 0.4, 0.1, 0.1,0.05, 2)

ml model lf beauty_mixture /p1 /p2 /p3 /p4 /p5 /sig
ml init start, copy
ml maximize
nlcom p0: 1-_b[p1:_cons]-_b[p2:_cons]-_b[p3:_cons]-_b[p4:_cons]-_b[p5:_cons]

drop postp*
```

Use Delta Method to obtain p_0

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
p0	.3026407	.029052	10.42	0.000	.2456999	.3595815

30% are Level-0

$\hat{p_0} = 0.303\ (0.029)$

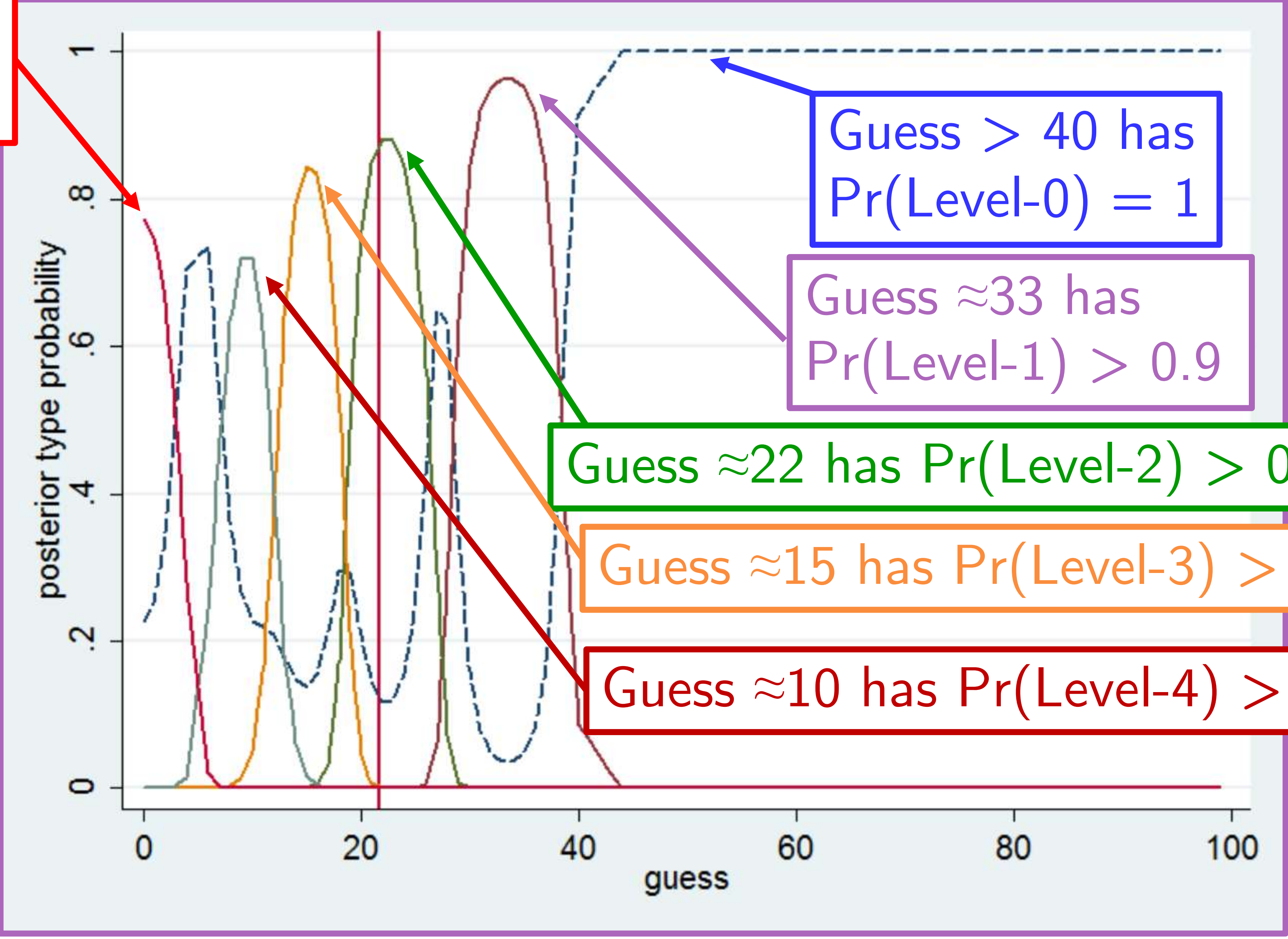
```
getmata postp5
sort y
line postp0 postp1 postp2 postp3 postp4 postp5 y , lpattern(- l l l l l)
```

Es... th...

- Estimate $p_1, p_2, p_3, p_4, p_5, \sigma$

```
mat star
ml model
ml init
ml maxim
nlcom p0
drop pos
getmata
getmata
getmata
sort y
line postp0 postp1 postp2 postp3 postp4 postp5 y , lpattern(- l l l l l)
```

Plot Posterior Type Probabilities



Guess ≈0 has Pr(Nash) > 0.75

Guess > 40 has Pr(Level-0) = 1

Guess ≈33 has Pr(Level-1) > 0.9

Guess ≈22 has Pr(Level-2) > 0.85

Guess ≈15 has Pr(Level-3) > 0.8

Guess ≈10 has Pr(Level-4) > 0.7

Estimating the

- Estimate
  $p_1,\ p_2,\ p_3,$
  $p_4,\ p_5,\ \sigma$

```
mat star

ml model
ml init
ml maxim
```

Guess in (0,10) has Pr(Level-0) > 0.7 (Level-5,6… Types Here? Need $J > 5$!)

```
getmata
```

Plot Posterior Type Probabilities

```
sort y
line postp0 postp1 postp2 postp3 postp4 postp5 y , lpattern(- l l l l l)
```

_cons]

Part II-plus: The Cognitive Hierarchy Model
第二部分加碼: 認知階層模型

Joseph Tao-yi Wang (王道一)
EEBGT, Experimetrics Module 6

The Cognitive Hierarchy (CH) Model

- Level-k model: Believe others exactly 1 level below themselves
- The Cognitive Hierarchy Model: Camerer (2003, 2004)
  - Population distribution over reasoning levels: Poisson(τ)

$$p(j) = \Pr(\text{Type} = j) = \left(\frac{e^{-\tau}\tau^j}{j!}\right), \; j = 0, 1, 2, \cdots$$

  - Type k believes others are Type 0, 1, ..., (k–1) with (upper) Truncated Poisson(τ):

$$p_k(j) = \Pr(\text{Type} = j|k) = \frac{\left(\frac{e^{-\tau}\tau^j}{j!}\right)}{\left(\sum_{m=0}^{k-1}\frac{e^{-\tau}\tau^m}{m!}\right)}, \; j = 0, \cdots, k-1$$

Cognitive Hierarchy Model of the Beauty Contest Game

- Type 1 Believe Others are Type 0 and Choose:
  - $b_1 = (2/3)[50] = 33.3$ as Type 0 Choose from Uniform[0,100]
- Type 2 Believe Others are Type 0 or 1 and Choose:
  - $b_2 = (2/3)[50 p_2(0) + b_1 p_2(1)]$
- Type 3 Believe Others are Types 0, 1 or 2 and Choose:
  - $b_3 = (2/3)[50 p_3(0) + b_1 p_3(1) + b_2 p_3(2)]$
- Type 4 Believe Others are Type 0, 1, 2 or 3 and Choose:
  - $b_4 = (2/3)[50 p_4(0) + b_1 p_4(1) + b_2 p_4(2) + b_3 p_4(3)]$

Cognitive Hierarchy Model of the Beauty Contest Game

- Type K has accurate beliefs about the society as $K \to \infty$
  - Not Nash (in general)!
  - Converges to SOPH if type distribution is indeed Poisson
- To Estimate the Cognitive Hierarchy Model:
  - Define Choice of each Type recursively
  - Assume Maximum Reasoning Levels = 4 for practical purposes
- Let Type 5 be Naïve Nash and Choose: $b_5 = 0$
  - Let Level-0 choose randomly from uniform distribution
    - First let's simulate some CH data

Simulating CH Data

- STATA: Simulate cog_hier_sim.dta

```
clear

set more off
set seed 9123456
set obs 500

egen i=fill(1/2)

* set "true" parameter values for simul

scalar tau=2.0
scalar sigma=2.0

*generate the computational error variable
```

```
*generate the computational error variable

gen e=sigma*rnormal()

*generate the level-of-reasoning for each individual,
*setting the maximum level to 5

gen level=rpoisson(tau)
replace level=5 if level>5

*generate the first few Poisson Probabilities;
*p5 is one minus the sum of the others.

scalar p0=exp(-tau)
scalar p1=p0*tau/1
scalar p2=-p1*tau/2
scalar p3-p2*tau/3
scalar p4=p3*tau/4

scalar p5=1-p0-p1-p2-p3-p4
```

1. Error $\sigma = 2$
2. Poisson $\tau = 2$

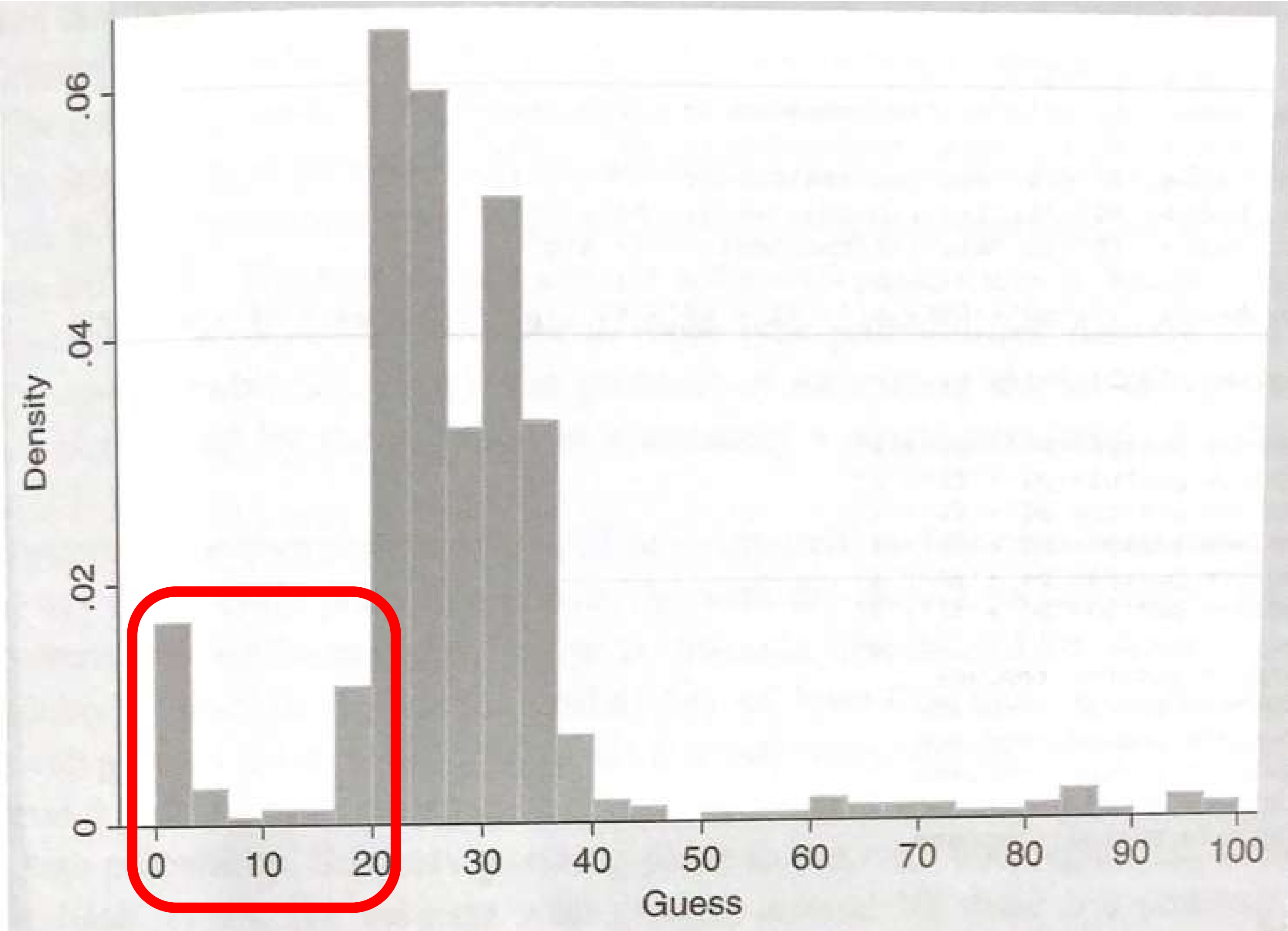
Simulating CH Data

- STATA: Simulate cog_hier_sim.dta
  - $\sigma = 2, \tau = 2$
- Type Predictions:
  - $b_0 = 50$
  - $b_1 = [50] \times (2/3) = 33$
  - $b_2 = [50 p_2(0) + b_1 p_2(1)] \times (2/3)$
  - $b_3 = [50 p_3(0) + b_1 p_3(1) + b_2 p_3(2)] \times (2/3)$; $b_5 = 0$ (Naïve Nash)
  - $b_4 = [50 p_4(0) + b_1 p_4(1) + b_2 p_4(2) + b_3 p_4(3)] \times (2/3)$

```
* generate the "best guesses" for each level of reasoning;
* Notethat type 5 is "naive Nash" with best-guess zero.

scalar b0=50
scalar b1=.67*b0
scalar b2=.67*(pl*b1+p0*b0)/ (p1+p0)
scalar b3=.67*(p2*b2+pl*b1+p0*b0)/ (p2+p1+p0)
scalar b4=.67 (p3+b3+p2*b2+p1*b1+p0*b0)/ (p3+p2+pl+p0)
scalar b5=0

* generate the guesses

gen y=round((level==0)*100+uniform()+(level==1)*(blte) ///
       +(level==2)+(b2+e)+(level==3)*(b3+e)  ///
         +(level==4)*(b4+e)+(level==5)*abs(0+e),1)
hist y, bin(30) xtitle(guess)
```

Simulated Data for the p-Beauty Contest Game

- Very few guesses below 20 (except 0)
  - Because most people have type $= 0, 1, 2, 3$
- Best guess converges to a lower bound
  - Here, lower bound $= 20$
  - Guesses < 20 can only be explain by type 0!



Estimating the Cognitive Hierarchy Model

- Type j : $y|_{\text{Type } j} = b_j + \epsilon$, $\epsilon \sim N(0, \sigma^2)$, $j = 1,\ldots,J = 5$
  - Conditional Density Functions:
  - Level-0: $f(y|T_0) = 1/100,\ 0 \leq y \leq 100$
  - Level-j: $f(y|T_j) = \frac{1}{\sigma}\phi\left(\frac{y - b_j}{\sigma}\right),\ 0 \leq y \leq 100 \quad (j = 1,\ldots,J\ )$
- Sample Log-Likelihood with Mixture $p(0)$, $p(1)$,..., $p(5)$:
  - For y_i, $i = 1,\ldots,n$:
  - Error σ
  - $p(k)$: Poisson(τ)

$$\log L = \sum_{i=1}^{n} \ln \left[\frac{p(0)}{100} + \sum_{j=1}^{J} p(j) \frac{1}{\sigma} \phi\left(\frac{y_i - b_j}{\sigma}\right) \right]$$

Estimat

```
program drop _all

*Log-likelihood evaluation program (ch) starts here

program define cog_heir
args logl sig tau
tempvar f0 f1 f2 f3 f4 f5 l
tempname p0 p1 p2 p3 p4 p5 b0 b1 b2 b3 b4 b5

scalar `p0'=exp(-`tau')
scalar `p1'=`p0'*`tau'/1
scalar `p2'=`p1'*`tau'/2
scalar `p3'=`p2'*`tau'/3
scalar `p4'=`p3'*`tau'/4
scalar `p5'=1-`p0'-`p1'-`p2'-`p3'-`p4'

scalar `b0'=50
scalar `b1'=.67*`b0'
scalar `b2'=.67*(`p1'*`b1'+`p0'*`b0')/(`p1'+`p0')
scalar `b3'=.67*(`p2'*`b2'+`p1'*`b1'+`p0'*`b0')/(`p2'+`p1'+`p0')
scalar `b4'=.67*(`p3'*`b3'+`p2'*`b2'+`p1'*`b1'+`p0'*`b0')/(`p3'+`p2'+`p1'+`p0')
```

- STATA Code to estimate:
  1. computational error parameter σ
  2. Poisson mean τ

Estimating CH

- STATA Code to estimate:
  1. computational error parameter σ
  2. Poisson mean τ

```
quietly{

gen double `f0'=0.01
gen double `f1'=(1/`sig')*normalden((y-`b1')/`sig')
gen double `f2'=(1/`sig')*normalden((y-`b2')/`sig')
gen double `f3'=(1/`sig')*normalden((y-`b3')/`sig')
gen double `f4'=(1/`sig')*normalden((y-`b4')/`sig')
gen double `f5'=(1/`sig')*normalden((y-0)/`sig')

gen double `l'=`p0'*`f0'+`p1'*`f1'+`p2'*`f2'+`p3'*`f3'+`p4'*`f4'+`p5'*`f5'

replace `logl'=ln(`l')
```

```
replace `logl'=ln(`l')

replace postp0=`p0'*`f0'/`l'
replace postp1=`p1'*`f1'/`l'
replace postp2=`p2'*`f2'/`l'
replace postp3=`p3'*`f3'/`l'
replace postp4=`p4'*`f4'/`l'
replace postp5=`p5'*`f5'/`l'

putmata postp0, replace
 putmata postp1, replace
 putmata postp2, replace
 putmata postp3, replace
 putmata postp4, replace
 putmata postp5, replace
}

end
```

```
* create posterior prob variables, set starting values and call ML program (ch)

gen postp0=.
gen postp1=.
gen postp2=.
gen postp3=.
gen postp4=.
gen postp5=.

mat start=( 2,2)
ml model lf cog_heir /sig /tau
ml init start, copy

ml maximize

drop postp*

getmata postp0
getmata postp1
getmata postp2
getmata postp3
getmata postp4
getmata postp5
```

```
getmata postp5

sort y

line postp0 postp1 postp2 postp3 postp4 postp5 y , lpattern(- l l l l l) ///
legend(off) xlabel(0(10)100) xtitle(guess) ytitle("posterior type probability")
```

- STATA Code to estimate:
  1. computational error parameter σ
  2. Poisson mean τ

Estimating the Cognitive Hierarchy Model

- Computational error parameter $\sigma = 1.998$
- Poisson mean $\tau = 2.028$ (instead of type probabilities)

```
                                                  Number of obs     =        500
                                                  Wald chi2(0)      =          .
Log likelihood = -1746.3042                       Prob > chi2       =          .

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         sig |
       _cons |   1.998404   .1071528    18.65   0.000     1.788388    2.208419
-------------+----------------------------------------------------------------
         tau |
       _cons |   2.028311   .0503448    40.29   0.000     1.929637    2.126985
------------------------------------------------------------------------------
```

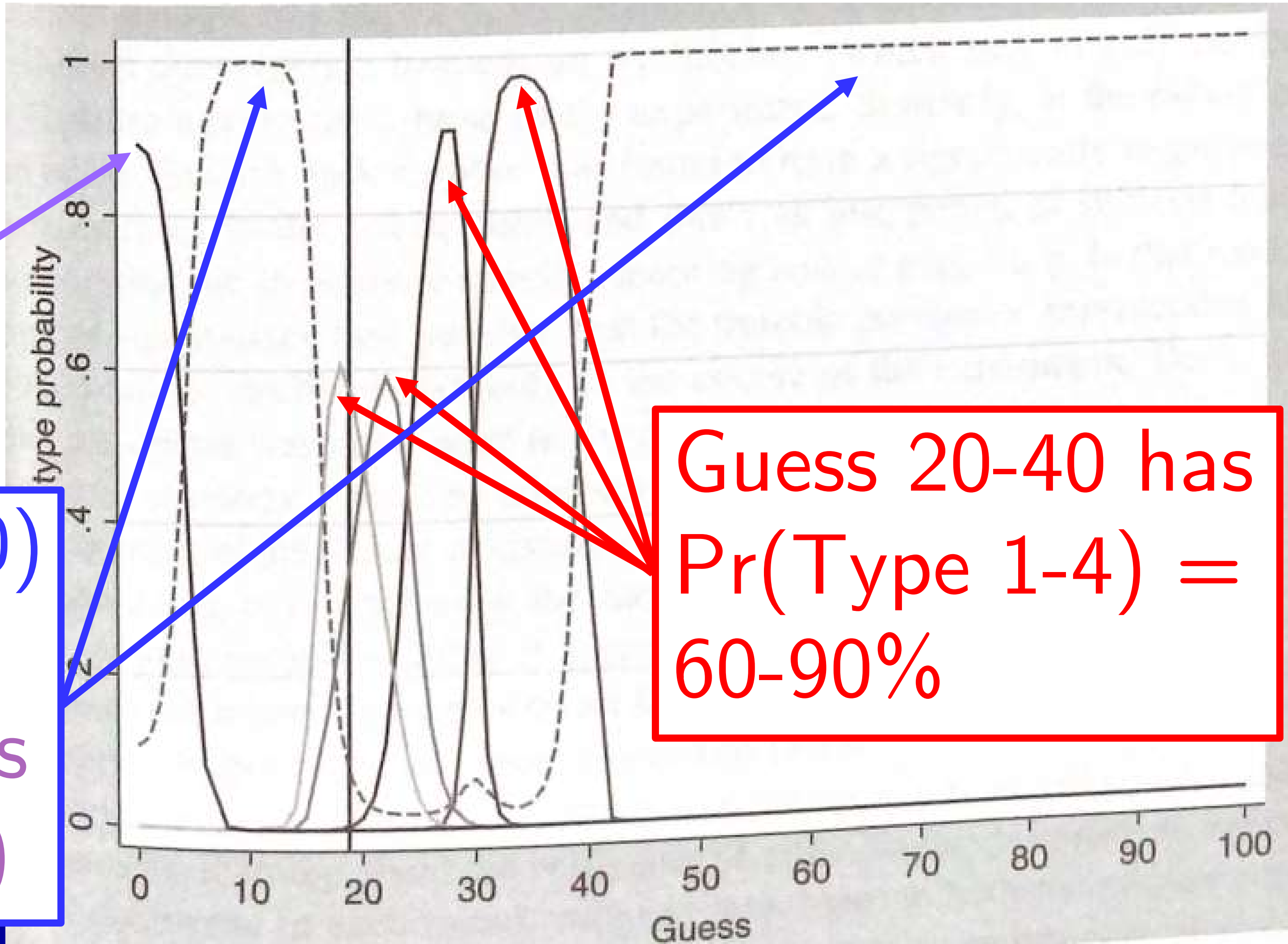
Estimat

```
sort y

line postp0 postp1 postp2 postp3 postp4 postp5 y , lpattern(- l l l l l) ///
legend(off) xlabel(0(10)100) xtitle(guess) ytitle("posterior type probability")
```

- Ploting posterior type probabilities

Guess ≈0 has Pr(Type 5) = 90%

Guess > 40 has Pr(Type 0) = 100%, as well as 10-15 (These are winning guesses low τ CH cannot capture!)



Guess 20-40 has Pr(Type 1-4) = 60-90%

Part III: A Public Goods Game Experiment
第三部分: 公共財自願捐獻賽局實驗

Joseph Tao-yi Wang (王道一)
EEBGT, Experimetrics Module 6

Public Goods Game Experiment

- n $(= 7)$ Subjects per group with endowment e_i $(= 10)$
  - Contribute to Public Account (or keep in Private Account)
  - MPCR $= k/n$: Public Account multiplied by k, but divided equally between all n members
- Doubly Censored Data: Contribute between 0 and e_i
  - Use Two-Limit Tobit Model (Nelson, 1976)
- Unique Nash Equilibrium: Zero Contribution
  - Experimental Data: Some positive contributions
  - Bardsley (2000): Uncover Motivations Behind Them

Bardsley (2000): Why Contribution Decreases?

1. Learning to be Rational (learn incentive structure)
2. ~~Social Learning (learn about others' behavior)~~

- Bardsley (2000): Conditional Information Lottery (CIL)
  - Play 1 Real Round mixed with 19 Fake Rounds against Computer, but only pay the real round
    - Subjects treat each round as real, but past rounds are not informative: They are fake if this round is real!
- Bardsley (2000): Take Turns to Contribute
  - See Previous Contributions Before Contributing

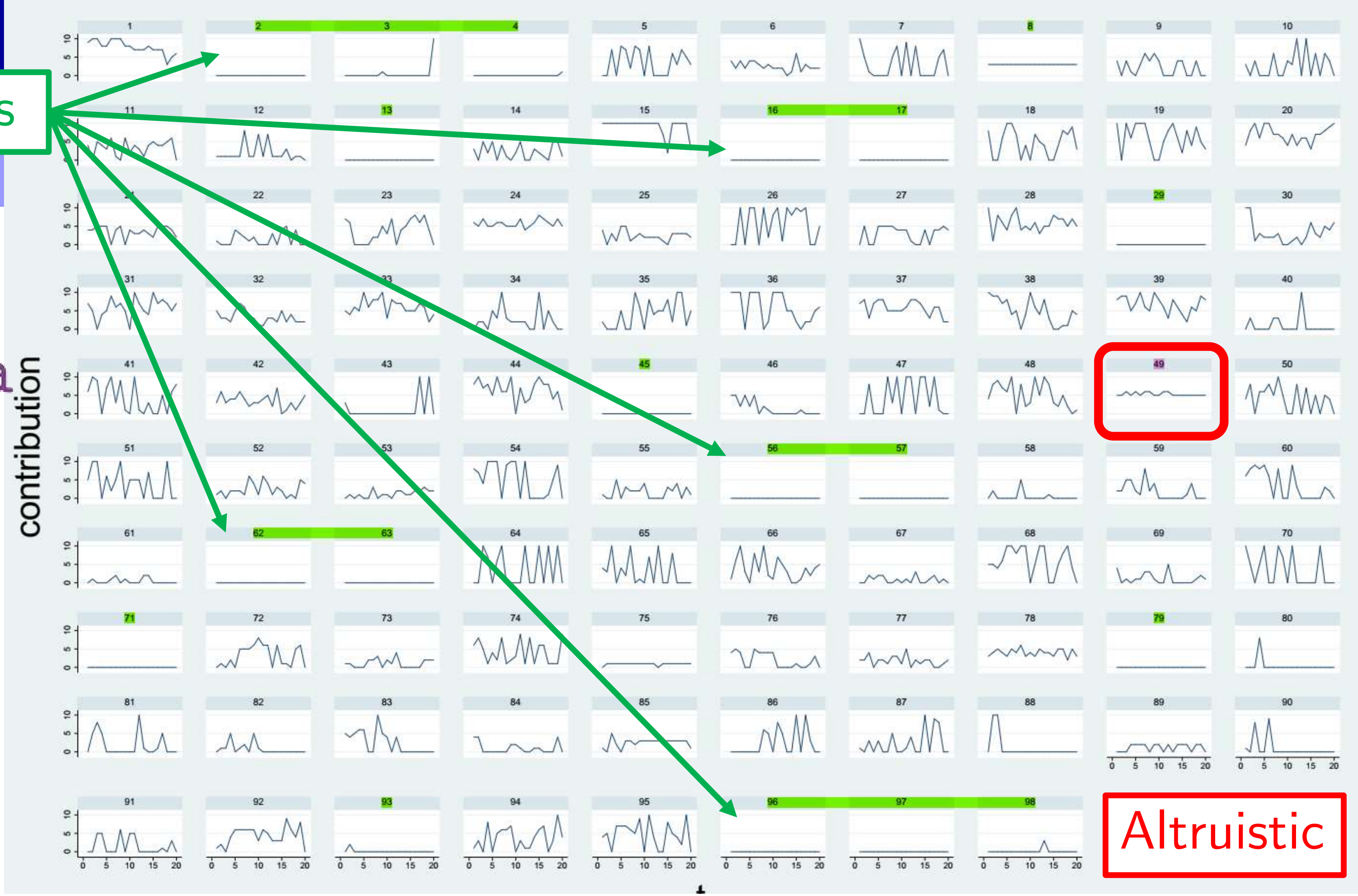
Bardsley (2000): Take Turns to Contribute

- See Previous Contributions Before Contributing
- Use Mixture Model to Address Different Motivations:

1. Reciprocator (Depends on Previous Contributions)
    - Contributes if Median of Previous Contribution is High
2. Strategist (Depends on Position in Sequence)
    - Contributes to Induce Later Contributions
3. Free-Rider
    - Contributes 0 Regardless

The Data

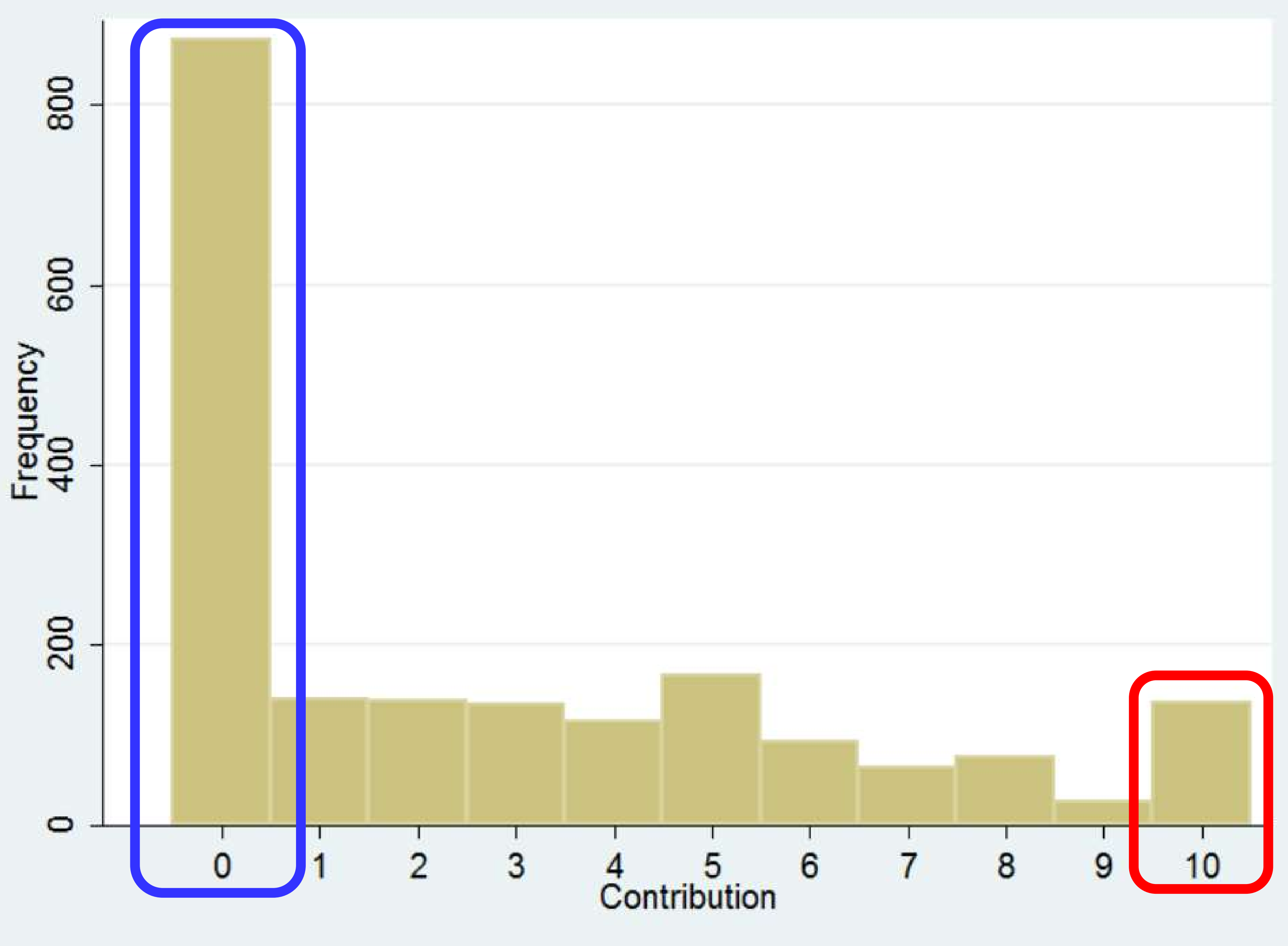
- Data (n=98)
  `bardsley.dta`
- STATA Command:
  `xtset i t`
  `xtline y`



- Results:
  - Do it to all panel data to catch Between-Subject Heterogeneity

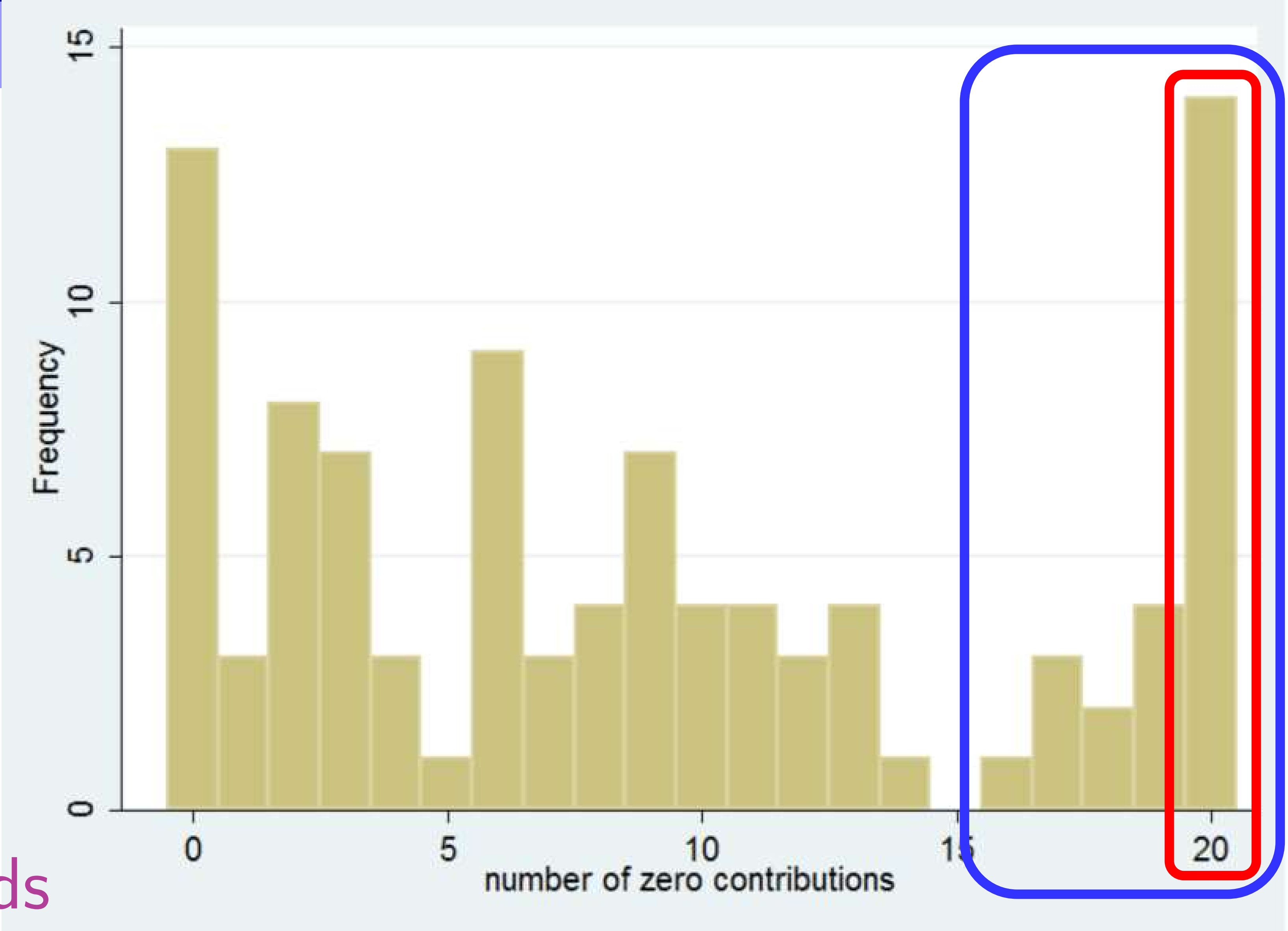
The Data

- Data (n=98)
  `bardsley.dta`
- STATA Command:
  `hist y`
- Results:
  - Many censored at 0
  - Some censored at 10
  - Mean = 2.711
  - Median = 1.0

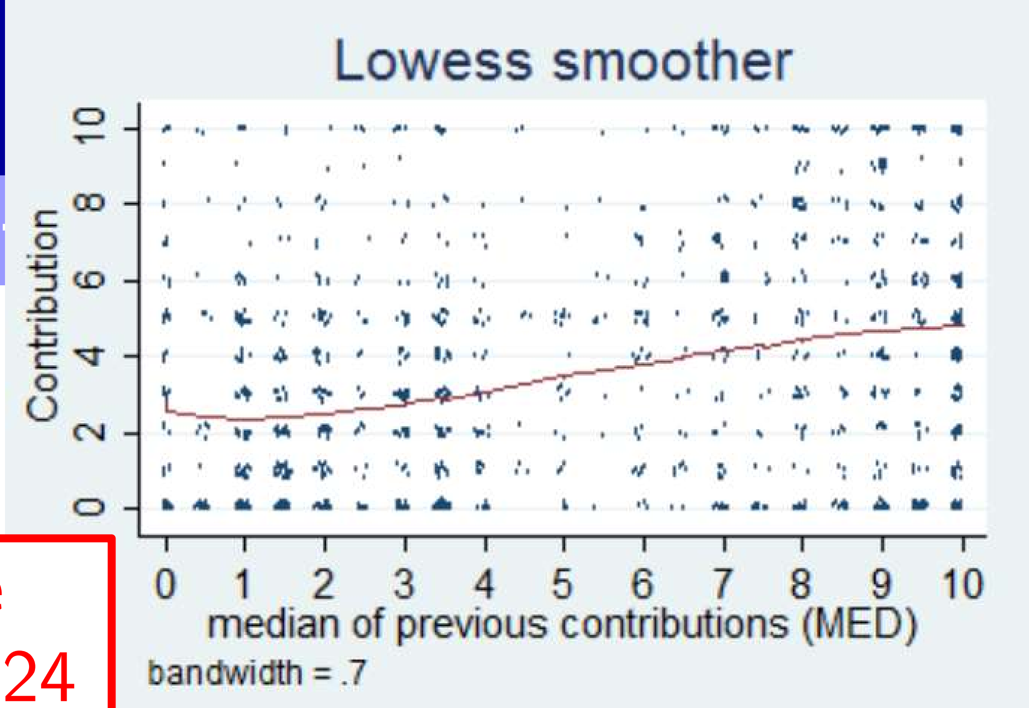


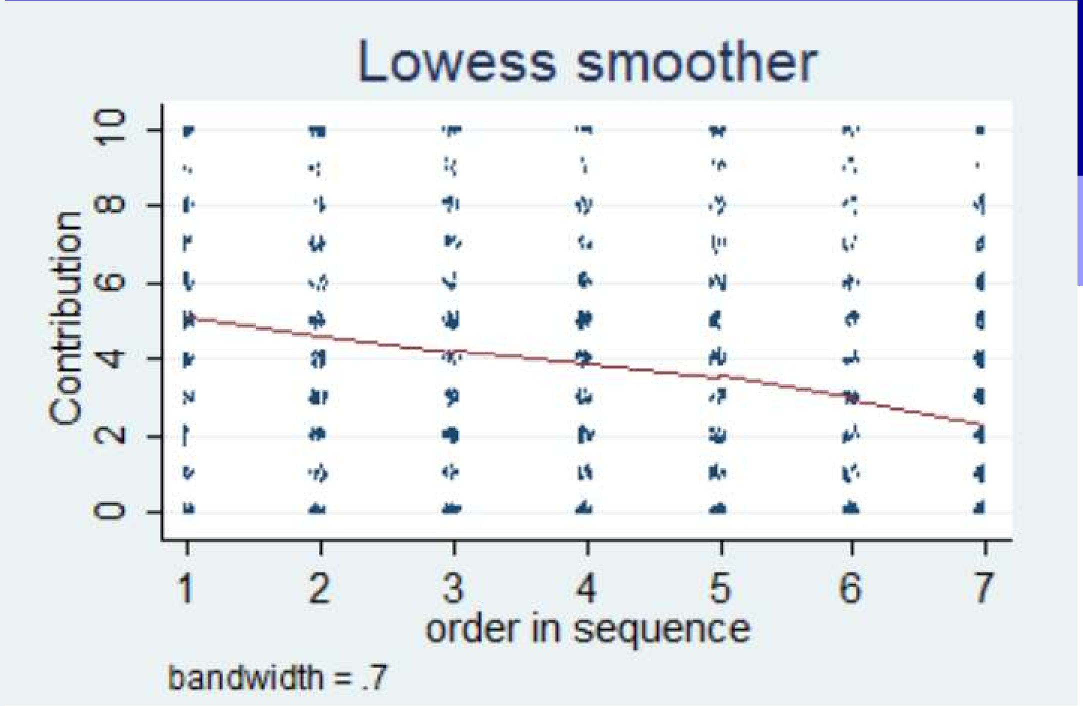
The Data

- Data (n=98)
  `bardsley.dta`
- STATA Command:
  `hist y=0 ?`
- Results:
  - Identify Free Riders
  - 14.3% always give 0
  - 24.5% mostly give 0 in 16 out of 20 rounds

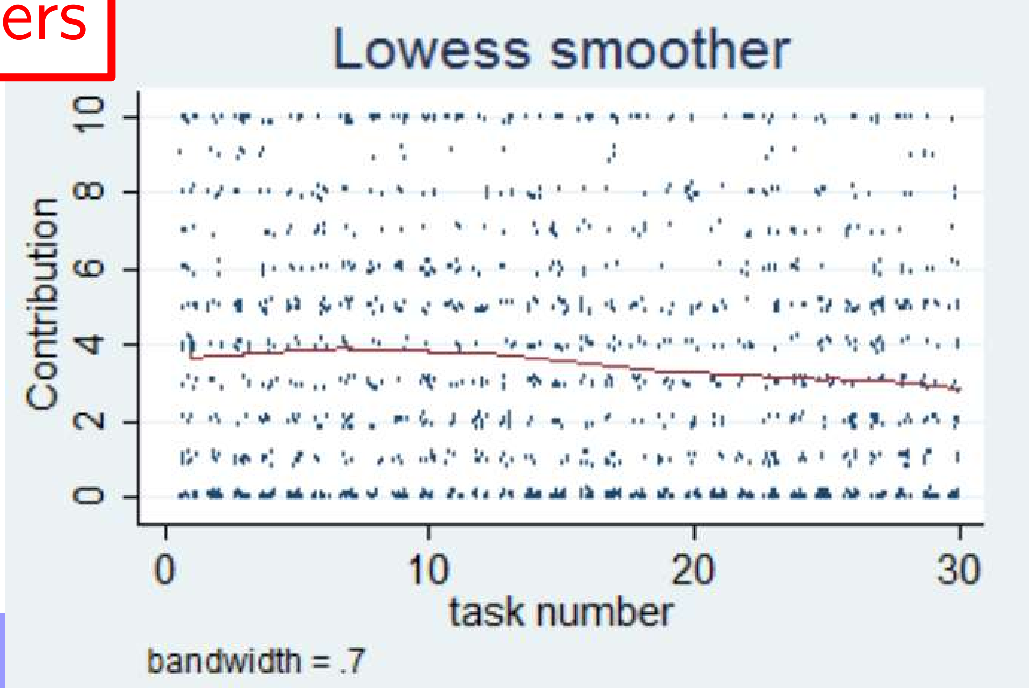


The Da





After we Exclude 24 Free-Riders



- Contribute more if:
  1. Higher MED (Median of Previous Contribution)
  2. Earlier Order in Sequence
  3. Earlier Task Number
  - All Monotonic/Linear!

Finite Mixture 2-Limit Tobit Model with Tremble

- Bardsley and Moffatt (2007)
- Observe n Subjects for T tasks
  - Either Reciprocator, Strategist and Free-Rider for all T tasks
- Subject i contributes y_{it} in task t between 0 and 10
- 2-Limit Tobit Model for Reciprocator and Strategist:

$$y_{it} = \begin{cases} 0 & \text{if } y_{it}^* \leq 0 \\ y_{it}^* & \text{if } 0 < y_{it}^* < 10 \\ 10 & \text{if } y_{it}^* \geq 10 \end{cases}$$

(Regime 1: No Contribution At All)
(Regime 2: Contribute b/w 0-10)
(Regime 3: Full Contribution of 10)

Actual

Desired

Finite Mixture 2-Limit Tobit Model with Tremble

- Desired Contribution of Subjects $i = 1\text{-}n$ in tasks $t = 1\text{-}T$ are
- Reciprocator (rec) Median of Previous Contributions

$$y^*_{it} = \beta_{10} + \beta_{11} MED_{it} + \beta_{13}(TSK_{it} - 1) + \epsilon_{it,rec}$$

Desired

>0 for Reciprocity

<0: Learning

$$\epsilon_{it,rec} \sim N(0, \sigma_1^2)$$

- Strategist (str) Decision Order

Minus 1

Task Number (1-30)

$$y^*_{it} = \beta_{20} + \beta_{22}(ORD_{it} - 1) + \beta_{23}(TSK_{it} - 1) + \epsilon_{it,str}$$

E(Contribution | Task 1, Order 1)

<0 for Strategic Behavior

$$\epsilon_{it,str} \sim N(0, \sigma_2^2)$$

- Free-Rider (fr): None $y_{it} = 0$

Finite Mixture 2-Limit Tobit Model with Tremble

- Prior Expectation of Others' Contribution
  - Set $MED = 8.00$ if $ORD = 1$ (trial-and-error to max. log-L)
- Mistakes (Moffatt and Peters, 2001): Tremble ω
  - Decreasing magnitude over time $\omega_{it} = \omega_0 \exp\left[\omega_1 (TSK_{it} - 1)\right]$
  - Initial tremble probability ω_0 vs. rate of decay $\omega_1 < 0$
- Regime 1 ($y = 0$)
- Regime 2 ($0 < y < 10$)
- Regime 3 ($y = 10$)

Finite Mixture 2-Limit Tobit Model with Tremble

- Regime 1 ($y = 0$):

Tremble: 0-10 with Equal Chance

  - $\Pr(y_{it} = 0 | i = \text{rec}) =$
  $$(1-\omega_{it})\Phi\left(\frac{-\beta_{10} - \beta_{11} MED_{it} - \beta_{13}(TSK_{it} - 1)}{\sigma_1}\right) + \frac{\omega_{it}}{11}$$
  - $\Pr(y_{it} = 0 | i = \text{str}) =$
  $$(1-\omega_{it})\Phi\left(\frac{-\beta_{20} - \beta_{22}(ORD_{it} - 1) - \beta_{23}(TSK_{it} - 1)}{\sigma_2}\right) + \frac{\omega_{it}}{11}$$
  - $\Pr(y_{it} = 0 | i = \text{fr}) = 1 - \dfrac{10\omega_{it}}{11}$

Finite Mixture 2-Limit Tobit Model with Tremble

- Regime 2 ($0 < y < 10$):

Tremble: Uniform[-0.5, 10.5]

  - $f(y_{it}|i = \text{rec}) =$

$$(1-\omega_{it})\frac{1}{\sigma_1}\Phi\left(\frac{y_{it}-\beta_{10}-\beta_{11}MED_{it}-\beta_{13}(TSK_{it}-1)}{\sigma_1}\right)+\frac{\omega_{it}}{11}$$

  - $f(y_{it}|i = \text{str}) =$

$$(1-\omega_{it})\frac{1}{\sigma_2}\Phi\left(\frac{y_{it}-\beta_{20}-\beta_{22}(ORD_{it}-1)-\beta_{23}(TSK_{it}-1)}{\sigma_2}\right)+\frac{\omega_{it}}{11}$$

  - $f(y_{it}|i = \text{fr}) = \frac{\omega_{it}}{11}$

Finite Mixture 2-Limit Tobit Model with Tremble

Tremble: 0-10 with Equal Chance

- Regime 3 ($y = 10$):
  - $\Pr(y_{it} = 10 | i = \text{rec}) =$

$$(1-\omega_{it})\left[1-\Phi\left(\frac{10-\beta_{10}-\beta_{11}MED_{it}-\beta_{13}(TSK_{it}-1)}{\sigma_1}\right)\right]+\frac{\omega_{it}}{11}$$

  - $\Pr(y_{it} = 10 | i = \text{str}) =$

$$(1-\omega_{it})\left[1-\Phi\left(\frac{10-\beta_{20}-\beta_{22}(ORD_{it}-1)-\beta_{23}(TSK_{it}-1)}{\sigma_2}\right)\right]+\frac{\omega_{it}}{11}$$

  - $\Pr(y_{it} = 10 | i = \text{fr}) = \dfrac{\omega_{it}}{11}$

Finite Mixture 2-Limit Tobit Model with Tremble

- Likelihood Function is L_i

$$= p_{\text{rec}} \prod_{t=1}^{T} \Pr(y_{it} = 0|\text{rec})^{I_{y_{it}=0}} f(y_{it}|\text{rec})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{rec})^{I_{y_{it}=10}}$$

$$+ p_{\text{str}} \prod_{t=1}^{T} \Pr(y_{it} = 0|\text{str})^{I_{y_{it}=0}} f(y_{it}|\text{str})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{str})^{I_{y_{it}=10}}$$

$$+ p_{\text{fr}} \prod_{t=1}^{T} \Pr(y_{it} = 0|\text{fr})^{I_{y_{it}=0}} f(y_{it}|\text{fr})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{fr})^{I_{y_{it}=10}}$$

- $\hat{\beta}_{10}, \ldots, \hat{\beta}_{23}, \hat{\sigma}_1, \hat{\sigma}_2; \hat{\omega}_0, \hat{\omega}_1; \hat{p}_{\text{rect}}, \hat{p}_{\text{str}}, \hat{p}_{\text{fr}}$ maximize $\log L = \sum_{i=1}^{n} \log(L_i)$
  - (Sample Log-Likelihood)

STATA Code: Components of Log-Likelihood

- `p1_1,p2_1,p3_1`: $\Pr(y=0|\text{rec}), \Pr(y=0|\text{str}), \Pr(y=0|\text{fr})$
- `p1_2,p2_2,p3_2`: $f(y|\text{rec}), f(y|\text{str}), f(y|\text{fr}),\ 0<y<10$
- `p1_3,p2_3,p3_3`: $\Pr(y=10|\text{rec}), \Pr(y=10|\text{str}), \Pr(y=10|\text{fr})$
- `p1`:
  $\Pr(y_{it}=0|\text{rec})^{I_{y_{it}=0}} f(y_{it}|\text{rec})^{I_{0<y_{it}<10}} \Pr(y_{it}=10|\text{rec})^{I_{y_{it}=10}}$
- `p2`:
  $\Pr(y_{it}=0|\text{str})^{I_{y_{it}=0}} f(y_{it}|\text{str})^{I_{0<y_{it}<10}} \Pr(y_{it}=10|\text{str})^{I_{y_{it}=10}}$
- `p3`:
  $\Pr(y_{it}=0|\text{fr})^{I_{y_{it}=0}} f(y_{it}|\text{fr})^{I_{0<y_{it}<10}} \Pr(y_{it}=10|\text{fr})^{I_{y_{it}=10}}$

STATA Code: Components of Log-Likelihood

- `pp1`: $$\prod_{t=1}^{T} \Pr(y_{it} = 0|\text{rec})^{I_{y_{it}=0}} f(y_{it}|\text{rec})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{rec})^{I_{y_{it}=10}}$$
- `pp2`: $$\prod_{t=1}^{T} \Pr(y_{it} = 0|\text{str})^{I_{y_{it}=0}} f(y_{it}|\text{str})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{str})^{I_{y_{it}=10}}$$
- `pp3`: $$\prod_{t=1}^{T} \Pr(y_{it} = 0|\text{fr})^{I_{y_{it}=0}} f(y_{it}|\text{fr})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{fr})^{I_{y_{it}=10}}$$

STATA Code: Components of Log-Likelihood

- `theta1`: $\beta_{10}, \beta_{11}, \beta_{13}$
- `theta2`: $\beta_{20}, \beta_{22}, \beta_{23}$
- `sig1,sig2,w0,w1,w`: $\sigma_1, \sigma_2, \omega_0, \omega_1, \omega$
- `p_rec,p_str,p_fr`: $p_{\text{rect}}, p_{\text{str}}, p_{\text{fr}}$
- `pp,lnpp`: $L_i,\ LogL = \sum_{i=1}^{n} \log(L_i)$
- `postp1`: $\Pr(i = \text{rec}|y_{i1}, \ldots, y_{iT})$
- `postp2`: $\Pr(i = \text{str}|y_{i1}, \ldots, y_{iT})$
- `postp3`: $\Pr(i = \text{fr}|y_{i1}, \ldots, y_{iT})$

```
* ESTIMATION OF MIXTURE MODEL FOR BARDSLEY DATA

prog drop _all

* LIKELIHOOD EVALUATION PROGRAM STARTS HERE:

program define pg_mixture
args todo b lnpp
tempvar p1_1 p2_1 p3_1 p1_2 p2_2 p3_2 p1_3 p2_3 p3_3 p1 p2 p3 pp1 pp2 pp3 pp w

tempname theta1 theta2 sig1 sig2 w0 w1 p_rec p_str

* ASSIGN PARAMETER NAMES TO THE ELEMENTS OF THE PARAMETER VECTOR b:

mleval `theta1' = `b'  eq(1)
mleval `theta2' = `b'  eq(2)
mleval `sig1' = `b', eq(3) scalar
mleval `sig2'=`b', eq(4) scalar
mleval `w0'=`b', eq(5) scalar
mleval `w1'=`b', eq(6) scalar
mleval `p_rec'=`b', eq(7) scalar
mleval `p_str'=`b', eq(8) scalar
```

Local Variable: `theta1', `b', ...
vs. Global Variable: tsk_1 (below)

odel

2-Limit Tobit Model

```
mleval `p_rec'=`b', eq(7) scalar
mleval `p_str'=`b', eq(8) scalar

quietly{

* INITIALISE THE p* VARIABLES WITH MISSING VALUES:

gen double `p1_1'=.
gen double `p2_1'=.
gen double `p3_1'=.
gen double `p1_2'=.
gen double `p2_2'=.
gen double `p3_2'=.
gen double `p1_3'=.
gen double `p2_3'=.
gen double `p3_3'=.
gen double `p1'=.
gen double `p2'=.
gen double `p3'=.
gen double `pp1'=.
gen double `pp2'=.
gen double `pp3'=.
gen double `pp'=.
```

Local Variable: `theta1', `b', …
vs. Global Variable: tsk_1 (below)

```
* GENERATE THE TREMBLE PROBABILITY:

gen double `w'=`w0'*exp(`w1'*tsk_1)

* COMPUTE TYPE-CONDITIONAL DENSITIES UNDER REGIME 1:

replace `p1_1'=(1-`w')*normal(-`theta1'/`sig1')+`w'/11
replace `p2_1'=(1-`w')*normal(-`theta2'/`sig2')+`w'/11
replace `p3_1'=1-(10/11)*`w'

* COMPUTE TYPE-CONDITIONAL DENSITIES UNDER REGIME 2:

replace `p1_2'=(1-`w')*(1/`sig1')*normalden((y-`theta1')/`sig1')+`w'/11
replace `p2_2'=(1-`w')*(1/`sig2')*normalden((y-`theta2')/`sig2')+`w'/11
replace `p3_2'=`w'/11

* COMPUTE TYPE-CONDITIONAL DENSITIES UNDER REGIME 3:

replace `p1_3'=(1-`w')*(1-normal((10-`theta1')/`sig1'))+`w'/11
replace `p2_3'=(1-`w')*(1-normal((10-`theta2')/`sig2'))+`w'/11
replace `p3_3'=`w'/11

* MATCH TYPE-CONDITIONAL DENSITIES TO ACTUAL REGIMES (d IS REGIME):
```

Local Variable: `theta1', `b', ...
vs. Global Variable: tsk_1

...t Model

```
* MATCH TYPE-CONDITIONAL DENSITIES TO ACTUAL REGIMES (d IS REGIME):

replace `p1' = (d==1)*`p1_1'+(d==2)*`p1_2'+(d==3)*`p1_3'
replace `p2' = (d==1)*`p2_1'+(d==2)*`p2_2'+(d==3)*`p2_3'
replace `p3' = (d==1)*`p3_1'+(d==2)*`p3_2'+(d==3)*`p3_3'
```

$$\prod_{t=1} p_t \equiv \exp\left(\sum_t \ln p_t\right)$$

```
* FIND PRODUCT OF TYPE-CONDITIONAL DENSITIES FOR EACH SUBJECT:

by i: replace `pp1'=exp(sum(ln(max(`p1',1e-12))))
by i: replace `pp2'=exp(sum(ln(max(`p2',1e-12))))
by i: replace `pp3'=exp(sum(ln(max(`p3',1e-12))))
```

Sum $\ln(p_1)$ instead of product

Use "1e-12" if close to 0 to avoid negative infinity at ln(0)

```
* COMBINE TYPE-CONDITIONAL DENSITIES TO OBTAIN MARGINAL DENSITY FOR EACH SUBJECT
* (ONLY REQUIRED IN FINAL ROW FOR EACH SUBJECT):

replace `pp'=`p_rec'*`pp1'+`p_str'*`pp2'+(1-`p_rec'-`p_str')*`pp3'
replace `pp'=. if last~=1

* SPECIFY (LOG-LIKELIHOOD) FUNCTION WHOSE SUM OVER SUBJECTS IS TO BE MAXIMISED

mlsum `lnpp'=ln(`pp') if last==1

* GENERATE POSTERIOR TYPE PROBABILITIES, AND MAKE THESE AVAILABLE OUTSIDE THE PROGRAM
```

```
* GENERATE POSTERIOR TYPE PROBABILITIES, AND MAKE THESE AVAILABLE OUTSIDE THE PROGRAM

replace postp1=`p_rec'*`pp1'/`pp'
replace postp2=`p_str'*`pp2'/`pp'
replace postp3=(1-`p_rec'-`p_str')*`pp3'/`pp'
putmata postp1, replace
putmata postp2, replace
putmata postp3, replace
}
end

* END OF LOG-LIKELIHOOD EVALUATION PROGRAM

clear
set more off

* READ DATA

use 'bardsley'

by i: gen last=_n==_N

gen int d=1
```

Data: bardsley.dta

```
gen int d=1
replace d=2 if y>0
replace d=3 if y==10

gen double ord_1=ord-1
gen double tsk_1=tsk-1

* SET MEDIAN OF PREVIOUS CONTRIBUTIONS TO 8 FOR SUBJECTS IN FIRST POSITION:

replace med=8 if ord==1

* SPECIFY EXPLANATORY-VARIABLE LISTS FOR RECIPROCATOR (LIST1)
* AND STRATEGIST (LIST2) EQUATIONS:

local list1 "med tsk_1"
local list2 "ord_1 tsk_1"

* INITIALISE VARIABLES TO BE USED FOR POSTERIOR TYPE PROBABILITIES:

gen postp1=.
gen postp2=.
gen postp3=.

* SPECIFY STARTING VALUES
```

To make Constant = E(Contribution | Task 1, Order 1)

Set MED = 8 if ORD = 1 (trial-and-error to max. log-L)

Prior Expectation of Others' Contribution

```
* SPECIFY STARTING VALUES:

mat start=(0.57,-0.10,6.1,-0.93,-0.05,5.2,3.3,3.7,0.11,-0.05,0.26,0.49)

* SPECIFY LIKELIHOOD EVALUATOR, PROGRAM, AND PARAMETER NAMES:

ml model d0 pg_mixture (=`list1') (=`list2') /sig1 /sig2 /w0 /w1 /p1 /p2
ml init start, copy
```

Cannot use `lf` since mixture model has non-linear log-L

```
* USE ML COMMAND TO MAXIMISE LOG-LIKELIHOOD, AND STORE RESULTS AS "WITH_TREMBLE":

ml max, trace search(norescale)
est store with_tremble
```

Use D-Family: `d0` requires only log-L
(`d1`/`d2` requires analytical derivatives of log-L)

```
* COMPUTE THIRD MIXING PROPORTION USING DELTA METHOD:

nlcom p3: 1-[p1]_b[_cons]-[p2]_b[_cons]
```

Derive `p3` using the Delta Method!

```
* EXTRACT POSTERIOR TYPE PROBABILITIES AND PLOT THEM AGAINST
* NUMBER OF ZERO CONTRIBUTIONS:

drop postp1 postp2 postp3
```

Finite Mixtu

- STATA Results:

Log likelihood = -3267.6884

	Number of obs	=	1960
	Wald chi2(2)	=	108.07
	Prob > chi2	=	0.0000

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
eq1						
med	.598677	.0611812	9.79	0.000	.4787641	.7185899
tsk_1	-.0961739	.0202229	-4.76	0.000	-.13581	-.0565379
_cons	4.004374	.4541832	8.82	0.000	3.114192	4.894557
eq2						
ord_1	-.9644643	.0823741	-11.71	0.000	-1.125915	-.803014
tsk_1	-.0516766	.017189	-3.01	0.003	-.0853664	-.0179867
_cons	5.299353	.3828498	13.84	0.000	4.548981	6.049724
sig1						
_cons	3.442241	.1674649	20.56	0.000	3.114016	3.770466
sig2						
_cons	3.705603	.1611296	23.00	0.000	3.389794	4.021411

$\hat{\beta}_{11} = 0.599\ (0.061)$

$\hat{\beta}_{13} = -0.096\ (0.020)$

$\hat{\beta}_{10} = 4.004\ (0.454)$

$\hat{\beta}_{22} = -0.964\ (0.082)$

$\hat{\beta}_{23} = -0.052\ (0.017)$

$\hat{\beta}_{20} = 5.299\ (0.383)$

$\hat{\sigma}_1 = 3.442\ (0.167)$

$\hat{\sigma}_2 = 3.706\ (0.161)$

Finite Mixture 2-Limit Tobit Model with Tremble

- Reciprocator (rec)

$$y_{it}^* = \beta_{10} + \beta_{11} MED_{it} + \beta_{13}(TSK_{it} - 1) + \epsilon_{it,rec}$$

$$E(y^*|MED, TSK) = 4.004 + 0.599 MED - 0.096(TSK - 1)$$

>0 & <1 for Biased Reciprocity

<0: Learning

- Strategist (str)

$$y_{it}^* = \beta_{20} + \beta_{22}(ORD_{it} - 1) + \beta_{23}(TSK_{it} - 1) + \epsilon_{it,str}$$

$$E(y^*|ORD, TSK) = 5.299 - 0.964(ORD - 1) - 0.052(TSK - 1)$$

<0 for Strategic Behavior: First mover contribute 5.3, Last (ORD=7) contribute 0

Slower than Reciprocators

Finite Mixture 2-Limit Tobit Model with Tremble

- STATA Results:

$\hat{\omega}_0 = 0.104\ (0.032)$

$\hat{\omega}_1 = -0.049\ (0.022)$

$\hat{p}_{\text{rec}} = 0.271\ (0.048)$

$\hat{p}_{\text{str}} = 0.483\ (0.054)$

$\hat{p}_{\text{fr}} = 0.246\ (0.044)$

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
sig2						
_cons	3.705603	.1611296	23.00	0.000	3.389794	4.021411
w0						
_cons	.104174	.0321192	3.24	0.001	.0412216	.1671265
w1						
_cons	-.0492262	.0218191	-2.26	0.024	-.0919909	-.0064614
p1						
_cons	.2710853	.048467	5.59	0.000	.1760918	.3660788
p2						
_cons	.4832814	.0538021	8.98	0.000	.3778311	.5887316
p3	.2456333	.0436144	5.63	0.000	.1601506	.331116

STATA Code: Finite Mixture 2-Limit Tobit Model

```
* EXTRACT POSTERIOR TYPE PROBABILITIES AND PLOT THEM AGAINST
* NUMBER OF ZERO CONTRIBUTIONS:

drop postp1 postp2 postp3

getmata postp1
getmata postp2
getmata postp3

label variable postp1 "rec"
label variable postp2 "str"
label variable postp3 "fr"

by i: gen n_zero=sum(y==0)

scatter postp1 postp2 postp3 n_zero if last==1, title("with tremble") ///
ytitle("posterior probability") msymbol(x Dh Sh) jitter(3) saving(with, replace)
```

Plot posterior probabilities (with tremble)

STATA Code: Finite Mixture 2-Limit Tobit Model

```
* ESTIMATE MODEL WITHOUT TREMBLE, AND STORE RESULTS AS "WITHOUT_TREMBLE":

constraint 1 [w0]_b[_cons]=0.00
constraint 2 [w1]_b[_cons]=0.00

ml model d0 pg_mixture (=`list1') (=`list2') ///
/sig1 /sig2 /w0 /w1 /p1 /p2, constraints(1 2)

ml init start, copy
ml max, trace search(norescale)
est store without_tremble

nlcom p3: 1-[p1]_b[_cons]-[p2]_b[_cons]

* EXTRACT AND PLOT POSTERIOR TYPE PROBABILITIES FOR MODEL WITHOUT TREMBLE:

drop postp1 postp2 postp3
```

Estimate Restricted Model (without tremble)

```
* EXTRACT AND PLOT POSTERIOR TYPE PROBABILITIES FOR MODEL WITHOUT TREMBLE:

drop postp1 postp2 postp3

getmata postp1
getmata postp2
getmata postp3

label variable postp1 "rec"
label variable postp2 "str"
label variable postp3 "fr"

scatter postp1 postp2 postp3 n_zero if last==1, title("without tremble") ///
ytitle("posterior probability") msymbol(x Dh Sh) jitter(3) saving(without, replace)

* CARRY OUT LIKELIHOOD RATIO TEST FOR PRESENCE OF TREMBLE:

lrtest with_tremble without_tremble

* COMBINE THE TWO POSTERIOR PROBABILITY PLOTS

gr combine with.gph without.gph
```

Likelihood Ratio Test (with/without tremble)

Finite Mixture 2-Limit Tobit Model with Tremble

```
Likelihood-ratio test LR                                chi2(2)     =    149.89
(Assumption: without_trem~e nested in with_tremble)    Prob > chi2 =    0.0000
```

Results:

w0							
_cons						.0412216	.1671265
w1							
_cons						-.0919909	-.0064614
p1							
_cons		.0[illegible]				[illegible]0918	.3660788
p2							
_cons		.05[illegible]				[illegible]3311	.5887316
		Std.					
p3	.2456333	.0436144		5.63	0.000	.1601506	.331116

$\hat{\omega}_0 = 0.104\ (0.032)$ — Tremble starts at $\hat{\omega}_0 = 0.104$

$\hat{\omega}_1 = -0.049\ (0.022)$ — Decays to 0.041 by Task 20

$\hat{p}_{\text{rec}} = 0.271\ (0.048)$ — $\hat{p}_{\text{rec}} \approx 1/4$

$\hat{p}_{\text{str}} = 0.483\ (0.054)$ — $\hat{p}_{\text{str}} \approx 1/2$

$\hat{p}_{\text{fr}} = 0.246\ (0.044)$ — $\hat{p}_{\text{fr}} \approx 1/4$

24.5% mostly give 0 in 16 out of 20 rounds vs. 14.3% always give 0

No Tremble: $\hat{p}_{\text{fr}} = 0.143\ (0.035)$

Posterior Type Probabilities

- $\Pr(i = \text{rec}|y_{i1}, \ldots, y_{iT}) =$
$\frac{p_{\text{rec}}}{L_i} \prod_{t=1}^{T} \Pr(y_{it} = 0|\text{rec})^{I_{y_{it}=0}} f(y_{it}|\text{rec})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{rec})^{I_{y_{it}=10}}$

- $\Pr(i = \text{str}|y_{i1}, \ldots, y_{iT}) =$
$\frac{p_{\text{str}}}{L_i} \prod_{t=1}^{T} \Pr(y_{it} = 0|\text{str})^{I_{y_{it}=0}} f(y_{it}|\text{str})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{str})^{I_{y_{it}=10}}$

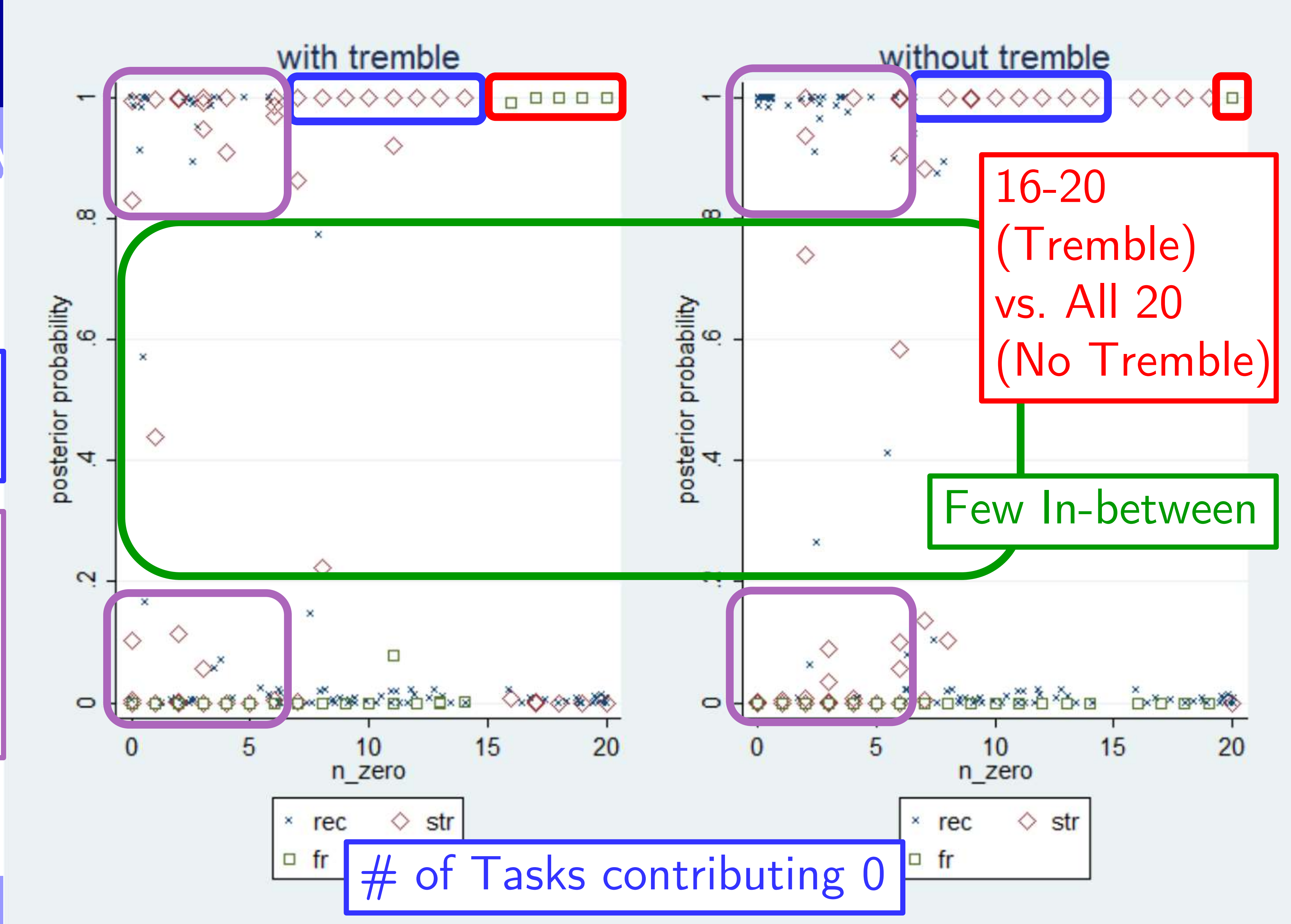
- $\Pr(i = \text{fr}|y_{i1}, \ldots, y_{iT}) =$
$\frac{p_{\text{fr}}}{L_i} \prod_{t=1}^{T} \Pr(y_{it} = 0|\text{fr})^{I_{y_{it}=0}} f(y_{it}|\text{fr})^{I_{0<y_{it}<10}} \Pr(y_{it} = 10|\text{fr})^{I_{y_{it}=10}}$

Posterior Ty

- STATA Results:

6-14(or 6-19) are Strategists

0-5 are Mixture of Strategists and Reciprocators



Conclusion: Finite Mixture Model

- Mixture Model accounts for Types in the Population
  - Infinite Mixture Model = Random Coefficient Model
- How it Works?
  - Economic Theory Predicts and Name Various Types
  - Construct Parametric Model for Behavior of Each Type
  - Estimated Using Population Data to Obtain:
    - Mixing Proportions and Parameters of Each Type
    - Individual Posterior Probability of being a Type

Acknowledgment

- This presentation is based on
  - Section 5.1-5 of the lecture notes of Experimetrics (and Section 17.3 of the textbook on Experimetrics),
- Prepared for a mini-course taught by Peter G. Moffatt (UEA) at National Taiwan University in Spring 2019
  - We would like to thank 康柏賢 for his in-class presentations