Repeated Game Play: Quantal Response Models
重複同一賽局: 手滑反應模型

Joseph Tao-yi Wang (王道一)
EE-BGT, Experimetrics Module 5

How Close Do People Play Mixed Nash Equilibrium?

- Observe Long Sequences of Repeated Play
  - Of the Same Subjects/Pairs/Group
  - Or, Do They Play Randomly?

- Quantal Response Equilibrium (QRE)
  - QR = Better Response, instead of Best Response (BR)
  - E = Correct Belief about Better Response of Others
  - (Simple) Log-QRE: One Noise parameter μ

Outline

- Pursue-Evade Game
- Two Non-Parametric Tests on Repeated Play
- QRE (Quantal Response Equilibrium)
  - Compute Choice Probabilities of QRE
  - Estimate QRE for Pursue-Evade Game
  - Estimate QRE for Pursue-Evade Game with Risk-Aversion
  - Apply QRE to Contest Game

Pursue-Evade Game (Rosenthal et al. 2003)

- Pursuer: L (left) or R (right)
- Evader: L (left) or R (right)
- Successful Evade
  - (L, R) or (R, L): no transfer
- Found
  - (L, L) or (R, R): $1 or $2 transferred
  - From Evader (lost) to Pursuer (won)

	L	R
L	1, -1	0, 0
R	0, 0	2, -2

Pursue-Evade Game (Rosenthal et al. 2003)

- Pure Nash Equilibrium (NE)?
  - BR for Pursuer
  - BR for Evader, so No pure Nash!
- Mixed NE?

Evader: p_{EL}, $1-p_{EL}$

Pursuer: p_{PL}, $1-p_{PL}$

	L	R
L	1, -1	0, 0
R	0, 0	2, -2

$$EV_P(L) = p_{EL} \cdot 1 + (1-p_{EL}) \cdot 0 = p_{EL}$$

$$= EV_P(R) = p_{EL} \cdot 0 + (1-p_{EL}) \cdot 2 = 2(1-p_{EL}) \Rightarrow p_{EL} = \frac{2}{3}$$

$$EV_E(L) = p_{PL} \cdot (-1) + (1-p_{PL}) \cdot 0 = -p_{PL}$$

$$= EV_E(R) = p_{PL} \cdot 0 + (1-p_{PL}) \cdot (-2) = -2(1-p_{PL}) \Rightarrow p_{PL} = \frac{2}{3}$$

Pursue-Evade Game (Rosenthal et al. 2003)

- Focus on pairs 21-34: Play 100 rounds
- In each pair in each round:
  - pur_L = 1 if Pursuer choose L
  - eva_L = 1 if Evader choose L
- Do people play Mixed NE?
  - Is L chosen 66.7% of the time?
- STATA command:
  - `table pair, contents(mean pur_L mean eva_L)`

	pair	period	pur_L	eva_L	pay
1	21	1	1	1	1
2	21	2	0	1	0
3	21	3	0	1	0
4	21	4	1	0	0
5	21	5	0	1	0
6	21	6	0	0	2
7	21	7	0	0	2
8	21	8	1	0	0
9	21	9	0	1	0
10	21	10	0	1	0
11	21	11	0	0	2
12	21	12	1	1	1
13	21	13	0	1	0
14	21	14	0	1	0
15	21	15	0	1	0
16	21	16	0	1	0
17	21	17	1	1	1
18	21	18	0	1	0

Pursue-Evade Game (Rosenthal et al. 2003)

- `table pair, contents(mean pur_L mean eva_L)`
- Is Pr(L) close to 2/3?
- Pairwise data show that
  - Pair 27, 28, 31, 32, 34 have Evaders with Pr(L) close to 2/3
  - Pr(L) is at most 0.67 for all but one Pursuer!
  - In aggregate…

Pair	mean(pur_L)	mean(eva_L)
21	.43	.84
22	.62	.59
23	.55	.59
24	.76	.78
25	.59	.86
26	.66	.82
27	.53	.67
28	.62	.7
29	.67	.78
30	.53	.59
31	.55	.69
32	.56	.69
33	.42	.55
34	.46	.66

Binomial Test for the Pursue-Evade Game

- sum pur_L eva_L

Variable	Obs	Mean	Std. Dev.	Min	Max
pur_L	1400	.5678571	.495551	0	1
eva_L	1400	.7007143	.4581088	0	1

- Evaders are on average closer to Nash than Pursuers

- Binomial Test for H_0: $p = 0.6667$ on Evaders
  - bitest eva_L=0.6667 if pair==31

Variable	N	Observed k	Expected k	Assumed p	Observed p
eva_L	100	69	66.67	0.66670	0.69000

```
Pr(k >= 69)            = 0.352782  (one-sided test)
Pr(k <= 69)            = 0.723211  (one-sided test)
Pr(k <= 64 or k >= 69) = 0.672191  (two-sided test)
```

Binomial Test for the Pursue-Evade Game

- sum pur_L eva_L

Variable	Obs	Mean	Std. Dev.	Min	Max
pur_L	1400	.5678571	.495551	0	1
eva_L	1400	.7007143	.4581088	0	1

- Evaders are on average closer to Nash than Pursuers

- Binomial Test for H_0: $p = 0.6667$ on Pursuers
  - bitest pur_L=0.6667 if pair==31

Variable	N	Observed k	Expected k	Assumed p	Observed p
pur_L	100	55	66.67	0.66670	0.55000

```
Pr(k >= 55)              = 0.994302  (one-sided test)
Pr(k <= 55)              = 0.009986  (one-sided test)
Pr(k <= 55 or k >= 79)   = 0.014760  (two-sided test)
```

Binomial Test for the Pursue-Evade Game

- Roughly half of pairs 21-34 are consistent with Nash!
  - * $p<0.05$
  - ** $p<0.01$

Pair	Pur_L	Eva_L	Binomial.Pur.	Binomial.Eva.	Runs.Pur.	Runs.Eva.
21	0.43	0.84	0.000**	0.000**	0.01*	0.67
22	0.62	0.59	0.340	0.112	0.05	0.74
23	0.55	0.59	0.015*	0.112	0.36	0.02*
24	0.76	0.78	0.056	0.019*	0.00**	0.09
25	0.59	0.86	0.112	0.000**	0.13	0.97
26	0.66	0.82	0.916	0.001**	0.00**	0.61
27	0.53	0.67	0.006**	1.000	0.01*	0.69
28	0.62	0.70	0.340	0.525	0.54	0.63
29	0.67	0.78	1.000	0.019*	0.10	0.62
30	0.53	0.59	0.006**	0.112	0.01*	0.90
31	0.55	0.69	0.015*	0.672	0.76	0.96
32	0.56	0.69	0.026*	0.672	0.09	0.01*
33	0.42	0.55	0.000**	0.015*	0.79	0.03*
34	0.46	0.66	0.000**	0.916	0.34	0.05

Runs Test (Siegel and Castellan, 1988)

- How many runs (same action repeated) in a sequence?
  - 0101101001 (Is 8 runs too many?) Negative serial correlation!
  - 0011111100 (Is 3 runs too few?) Positive serial correlation!
- For 100 runs:

Pursuer in pair 21:

1001000100010000100100101010101100010100101100100101011010111000010100
100010110001001100111011110010

(62 runs)

Evader in pair 21:

1110100011011111111101111111011110111110111110111110111111011111101111
110111111111111110111110111111

(29 runs)

Runs Test (Siegel and Castellan, 1988)

- Is the sequence generated by a fixed mixed strategy?
- For r run-sequence of N decisions
  - $N = m$ (L-choices) + n (R-choices) and m, $n > 20$
- Then, H_0: $r \sim N\left(\left(\frac{2mn}{N}+1\right), \frac{2mn(2mn-N)}{N^2(N-1)}\right)$
- Test Statistics: $z = \frac{r-\left(\frac{2mn}{N}+1\right)}{\sqrt{\frac{2mn(2mn-N)}{N^2(N-1)}}} \sim N(0,1)$

Runs Test (Siegel and Castellan, 1988)

Pursuer in pair 21:

1001000100010000100100101010110001010010110010010101101011100001010010001011000100110011101111001 0

(62 runs)

- $N = 100$, $m = 43$, $n = 57$, $r = 62$
- So, $z = \frac{r - \left(\frac{2mn}{N} + 1\right)}{\sqrt{\frac{2mn(2mn-N)}{N^2(N-1)}}} = \frac{62 - \left(\frac{2\times43\times57}{100} + 1\right)}{\sqrt{\frac{2\times43\times57(2\times43\times57-100)}{100^2(100-1)}}}$

$$= \frac{11.98}{\sqrt{23.78}} = 2.46$$

Runs Test (Siegel and Castellan, 1988)

- STATA: `runtest pur_L in 1/100, t(0.5)`

  Row 1 of the dataset — Threshold to separate runs; Can be any in (0,1)

- Result for Pursuer 21:

```
N(pur_L <= .5) = 57
N(pur_L >  .5) = 43
           obs = 100
       N(runs) = 62
             z = 2.46
      Prob>|z| = .01
```

- p-value = 0.01 (Not random!)

Positive Statistic

= More Runs than Expected

= Negative Serial Correlation

Runs Test (Siegel and Castellan, 1988)

- STATA: `runtest eva_L in 1/100, t(0.5)`
- Result for Evader 21:

```
N(eva_L <= .5) = 16
N(eva_L >  .5) = 84
           obs = 100
        N(runs) = 29
             z = .42
    Prob>|z| = .67
```

- Accept H_0 (random!)

```
Evader in pair 21:

1110100011011111111101111111011110111110111110111110111111011111101111
110111111111111101111101111111

(29 runs)
```

Runs Test (Siegel and Castellan, 1988)

- Most pairs in 21-34 are close to random sequences!
  - * p<0.05
  - ** p<0.01

Pair	Pur_L	Eva_L	Binomial.Pur.	Binomial.Eva.	Runs.Pur.	Runs.Eva.
21	0.43	0.84	0.000**	0.000**	0.01*	0.67
22	0.62	0.59	0.340	0.112	0.05	0.74
23	0.55	0.59	0.015*	0.112	0.36	0.02*
24	0.76	0.78	0.056	0.019*	0.00**	0.09
25	0.59	0.86	0.112	0.000**	0.13	0.97
26	0.66	0.82	0.916	0.001**	0.00**	0.61
27	0.53	0.67	0.006**	1.000	0.01*	0.69
28	0.62	0.70	0.340	0.525	0.54	0.63
29	0.67	0.78	1.000	0.019*	0.10	0.62
30	0.53	0.59	0.006**	0.112	0.01*	0.90
31	0.55	0.69	0.015*	0.672	0.76	0.96
32	0.56	0.69	0.026*	0.672	0.09	0.01*
33	0.42	0.55	0.000**	0.015*	0.79	0.03*
34	0.46	0.66	0.000**	0.916	0.34	0.05

Quantal Response Equilibrium (QRE)

- Nash = Consistent Beliefs + Best Response
- Mixed Nash Equilibrium is:

Evader: p_{EL} (L), $1-p_{EL}$ (R)

Pursuer	L	R
p_{PL} **L**	1, -1	0, 0
$1-p_{PL}$ **R**	0, 0	2, -2

$$EV_P(L) = p_{EL}\cdot 1 + (1-p_{EL})\cdot 0 = p_{EL}$$
$$= EV_P(R) = p_{EL}\cdot 0 + (1-p_{EL})\cdot 2 = 2(1-p_{EL}) \Rightarrow p_{EL} = \frac{2}{3}$$
$$EV_E(L) = p_{PL}\cdot(-1) + (1-p_{PL})\cdot 0 = -p_{PL}$$
$$= EV_E(R) = p_{PL}\cdot 0 + (1-p_{PL})\cdot(-2) = -2(1-p_{PL}) \Rightarrow p_{PL} = \frac{2}{3}$$

Quantal Response Equilibrium (QRE)

- Nash = Consistent Beliefs + Best Response
- QRE = Consistent Beliefs + Better Response
- Add stochastic term (noise) to expected payoffs
  - Mckelvey and Palfrey (1995)

1. $EV_P^*(L) = p_{EL} + \epsilon_{PL}$
2. $EV_P^*(R) = 2(1 - p_{EL}) + \epsilon_{PR}$
3. $EV_E^*(L) = -p_{PL} + \epsilon_{EL}$
4. $EV_E^*(R) = 2(1 - p_{PL}) + \epsilon_{ER}$

Evader: p_{EL} (L), $1 - p_{EL}$ (R)

Pursuer	L (p_{EL})	R ($1 - p_{EL}$)
p_{PL} **L**	1, -1	0, 0
$1 - p_{PL}$ **R**	0, 0	2, -2

Logit-QRE

- Let errors be Type-I Extreme Value Distribution (μ)

$$\epsilon_i \overset{\text{iid}}{\sim} F(\epsilon_i;\mu) = \exp\left(-\exp\left[-\frac{\epsilon_i}{\mu}\right]\right)$$

- Payoff Response Function (with Noise Parameter μ)
  - $\mu = 0$: No Noise (Mixed Nash)
  - $\mu = \infty$: All Noise (Random)

$$p_{PL} = \frac{\exp\left[\frac{p_{EL}}{\mu}\right]}{\exp\left[\frac{p_{EL}}{\mu}\right] + \exp\left[\frac{2(1-p_{EL})}{\mu}\right]}$$

$$p_{EL} = \frac{\exp\left[-\frac{p_{PL}}{\mu}\right]}{\exp\left[-\frac{p_{PL}}{\mu}\right] + \exp\left[-\frac{2(1-p_{PL})}{\mu}\right]}$$

Computing Logit-QRE using MATA (within STATA)

- To solve p_{PL}, p_{EL} for any given μ:

$$\min s_1 = \left[p_{PL} - \frac{\exp\left[\frac{p_{EL}}{\mu}\right]}{\exp\left[\frac{p_{EL}}{\mu}\right] + \exp\left[\frac{2(1-p_{EL})}{\mu}\right]}\right]^2$$

$$\min s_2 = \left[p_{EL} - \frac{\exp\left[-\frac{p_{PL}}{\mu}\right]}{\exp\left[-\frac{p_{PL}}{\mu}\right] + \exp\left[-\frac{2(1-p_{PL})}{\mu}\right]}\right]^2$$

  - Non-Linear!
  - We solve the Minimization Problem...

Computing Logit-

- STATA Code:

- Choose p to
- Minimize this

```
clear mata
set more off

* START MATA FROM WITHIN STATA

mata:

// SET STARTING VALUES FOR THE TWO PROBABILITIES

start=(0.5,0.5)

// CREATE PROGRAM ("vector_min") FOR EVALUATING 2x1 VECTOR (ss)
// WHOSE ELEMENTS ARE TO BE MINIMISED

void vector_min(todo, p, ss, S, H)
    {
 external mu
                 p_PL    = p[1]
                 p_EL    = p[2]

 s1= (p_PL-exp(p_EL/mu)/(exp(p_EL/mu)+exp(2*(1-p_EL)/mu)))^2
 s2= (p_EL-exp((-p_PL)/mu)/(exp((-p_PL)/mu)+exp(-2*(1-p_PL)/mu)))^2

                 ss = s1 \ s2

  }
```

Computing Logit-QRE using MATA (within STATA)

- STATA Code:

```
// BEGIN DEFINITION OF OPTIMISATION PROBLEM,
// RETURNING S, A PROBLEM-DESCRIPTION HANDLE CONTAINING DEFAULT VALUES

 S = optimize_init()

// MODIFY DEFAULTS

optimize_init_evaluator(S, &vector_min())
 optimize_init_evaluatortype(S, "gf0")
optimize_init_params(S, start)
 optimize_init_which(S,  "min" )
 optimize_init_tracelevel(S, "none")
optimize_init_conv_ptol(S, 1e-16)
 optimize_init_conv_vtol(S, 1e-16)

// RETURN TO STATA

end
```

Computing Logit-QRE using MATA (within STATA)

- STATA Code:

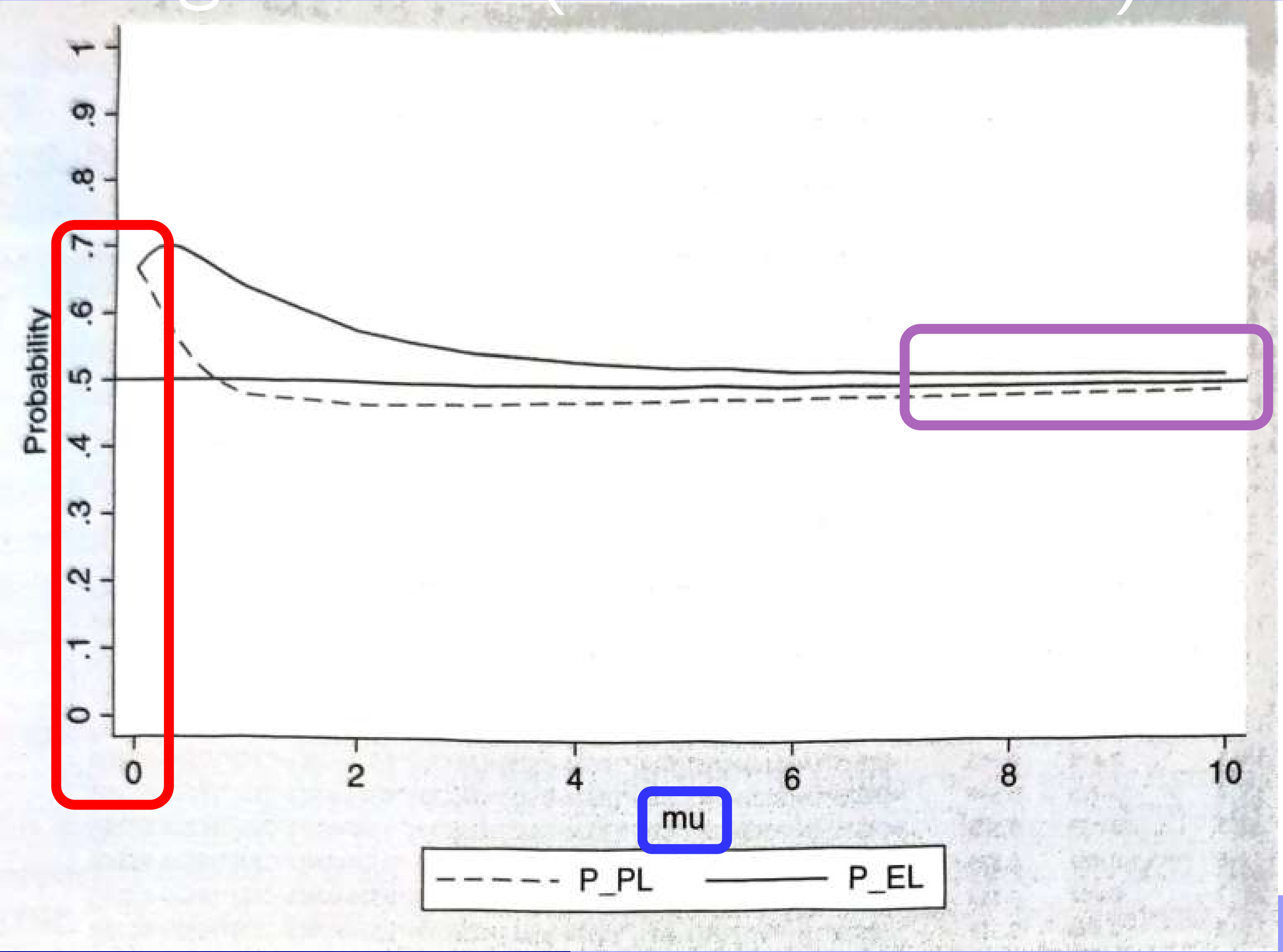
```
* SET VALUE OF mu (AND COPY STARTING PROBABILITIES AND mu INTO MATA)

scalar mu=.7
mata: mu=st_numscalar("mu")

* PERFORM OPTIMIZATION; STORE SOLUTION IN 2x1 VECTOR p:

mata: p = optimize(S)

* EXTRACT ELEMENTS OF p

mata: st_numscalar("p_PL",p[1])
mata:  st_numscalar("p_EL", p[2])

* DISPLAY RESULT

scalar list p_PL
scalar list p_EL
```

- Given μ
- Solve for p
  - In MATA
- And report:

```
. scalar list p_PL
      p_PL =   .50249676

. scalar list p_EL
      p_EL =   .66901177
```

Computing Logit-QRE using MATA (within STATA)

- Running over a range of μ we have:
  - $\mu = 0$: No Noise
    - Mixed Nash: 0.67
  - $\mu = \infty$: All Noise
    - Random at 0.5



Estimating Logit-QRE using STATA (by calling MATA)

- Given μ, we obtain two probabilities $p_{PL}(\mu)$, $p_{EL}(\mu)$
  - $y_{P,it}/y_{E,it} = 1$: Proposer/Evader in pair i choose L in Round t

$$\log L(\mu) = \sum_{i=1}^{n}\sum_{t=1}^{T} \ln\Big\{\Big[y_{P,it}p_{PL}(\mu) + (1-y_{P,it})\big(1-p_{PL}(\mu)\big)\Big] \times \Big[y_{E,it}p_{EL}(\mu) + (1-y_{E,it})\big(1-p_{EL}(\mu)\big)\Big]\Big\}$$

$$= n_{PL}\ln p_{PL}(\mu) + n_{PR}\big[1-\ln p_{PL}(\mu)\big] + n_{EL}\ln p_{EL}(\mu) + n_{ER}\big[1-\ln p_{EL}(\mu)\big]$$

  - n_{PL}/n_{PR} = # of times Pursuers choose L/R
  - n_{EL}/n_{ER} = # of times Evaders choose L/R

Sufficient statistics to estimate μ

Estimating Logit-QRE using STATA (by calling MATA)

- Calculate $\log L(\mu)$ for $n_{PL} = 500$ and $n_{EL} = 667$:

D9 | f_x | =(B$25*LN(B9)+LN(1-B9)*(B$24-B$25)+LN(C9)*B$26+LN(1-C9)*(B$24-B$26))

	A	B	C	D
1	mu	p_PL	p_EL	logl
2	0.001	0.6667	0.6667	-1388.3
3	0.1	0.641	0.686	-1371.7
4	0.2	0.611	0.697	-1356.8
5	0.3	0.582	0.7	-1345.6
6	0.4	0.556	0.697	-1337.8
7	0.5	0.534	0.689	-1332.9
8	0.6	0.516	0.68	-1330.3
9	0.7	0.502	0.669	-1329.4
10	0.8	0.492	0.658	-1329.7
11	0.9	0.484	0.647	-1330.8
12	1	0.478	0.638	-1332.2
13	2	0.466	0.575	-1349.5

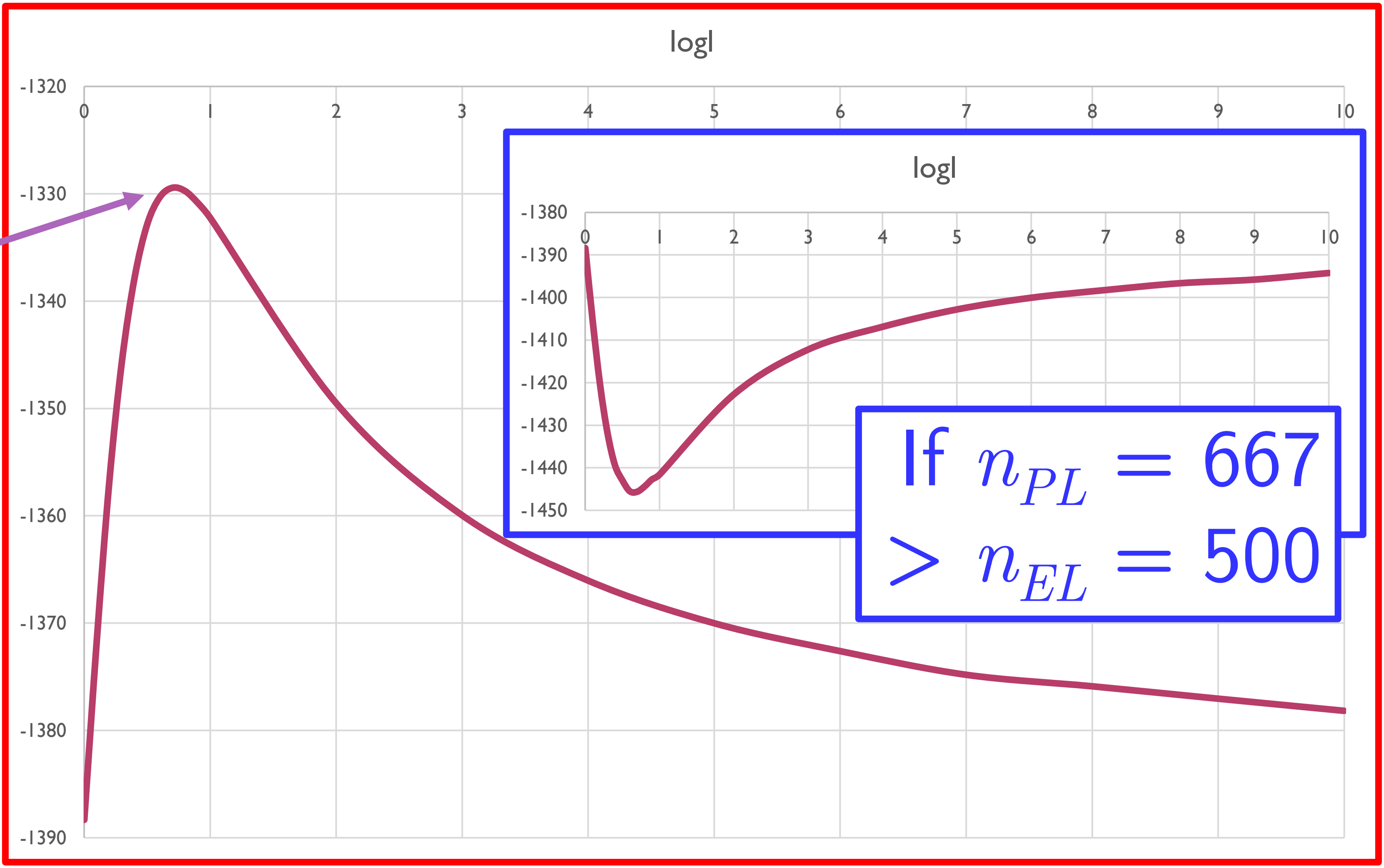
$$
\begin{aligned}
\log L(\mu) = n_{PL} \ln p_{PL}(\mu) \\
+ n_{PR}\left[1 - \ln p_{PL}(\mu)\right] \\
+ n_{EL} \ln p_{EL}(\mu) \\
+ n_{ER}\left[1 - \ln p_{EL}(\mu)\right]
\end{aligned}
$$

max $\log L$ at $\mu = 0.7$

13	2	0.466	0.575	-1349.5
	3	0.471	0.549	-1360
	4	0.475	0.536	-1366.1
	5	0.479	0.528	-1370
	6	0.482	0.523	-1372.6
	7	0.484	0.519	-1374.8
	8	0.486	0.517	-1375.9
20	9	0.487	0.515	-1377.1
21	10	0.489	0.513	-1378.2
22				
23				
24	n	1000		
25	n_PL	500		
26	n_EL	667		

Estimating Logit-QRE using STATA (by calling MATA)

- $\max\log L$ = -1329.43
- at $\mu = 0.72$ by ML
  - Near Excel's $\mu = 0.7$
- Limitations:
  1. $\log L$ may not be globally concave
     - Local maximum?
  2. Assume $n_{EL} > n_{PL}$
     - Minimum if not!!



Estimating Logit- … TA)

- STATA Code:
  - qre.do
- Simulate 1000:
  - $n_{PL} = 500$
  - $n_{EL} = 667$
- 2 initial vectors:
- MATA: `start`
  - ML: `sstart` (see next slide)

```
* GENERATE DATA SET

set obs 1000
gen int y_P = _n<501
gen int y_E = _n<668

* SET STARTING VALUES FOR COMPUTATION OF THE TWO PROBABILITIES

mat start=(0.67,0.67)

prog drop _all

* LOG-LIKELIHOOD EVALUATION PROGRAM (qre) STARTS HERE:

program define qre
 args lnf mu
 tempvar pp
 tempname p_PL p_EL

scalar mu=`mu'

* COPY STARTING PROBABILITIES AND mu INTO MATA

mata: start=st_matrix("start")
mata: mu=st_numscalar("mu")
```

Est… …ATA)

```
* PERFORM OPTIMIZATION; STORE SOLUTION IN 2x1 VECTOR p:

mata: p = optimize(S)

* EXTRACT ELEMENTS OF p

mata: st_numscalar("p_PL",p[1])
mata:  st_numscalar("p_EL", p[2])

* GENERATE JOINT PROBABILITY OF PURSUER'S AND EVADER'S DECISIONS:

 quietly gen double `pp'=((p_PL*y_P)+(1-p_PL)*(1-y_P))*((p_EL*y_E)+(1-p_EL)*(1-y_E))

* GENERATE LOG-LIKELIHOOD

 quietly replace `lnf'=ln(`pp')
 end

* SET STARTING VALUE FOR ML ROUTINE (NOTE: ONLY ONE PARAMETER, mu)

mat sstart=.6

* RUN ML

ml model lf qre ()
```

Calling MATA to solve $p_{PL}(\mu)$, $p_{EL}(\mu)$

Evaluates $\log L$ for a given μ

Pick μ yielding largest $\log L$

```
* RUN ML

ml model lf qre ()
ml init sstart, copy
ml maximize
```

Estimating Logit-QRE using STATA (by calling MATA)

- STATA Results:

```
. ml maximize

initial:       log likelihood = -1330.3224
rescale:       log likelihood = -1330.3224
Iteration 0:   log likelihood = -1330.3224
Iteration 1:   log likelihood = -1329.4369
Iteration 2:   log likelihood = -1329.4302
Iteration 3:   log likelihood = -1329.4302

                                                Number of obs     =      1,000
                                                Wald chi2(0)      =          .
Log likelihood = -1329.4302                     Prob > chi2       =          .
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   .7197914   .0974524     7.39   0.000     .5287882    .9107947
------------------------------------------------------------------------------
```

We find $\mu = 0.72$ achieving maximum $\log L =$ -1329.4302

Risk-Averse Logit-QRE Model

- Risk-Averse Nash Equilibrium with CRRA: **Evader**
  - Constant Relative Risk Aversion (r = RRA):

$$U(x) = \frac{x^{1-r}}{1-r}, x \geq 0$$

$$= -\frac{(-x)^{1-r}}{1-r}, x < 0$$

Pursuer

		L (p_{EL})	R ($1-p_{EL}$)
p_{PL}	L	1, -1	0, 0
$1-p_{PL}$	R	0, 0	2, -2

$$EU_P(L) = p_{EL} \cdot \frac{1}{1-r} + (1-p_{EL}) \cdot 0 = \frac{p_{EL}}{1-r}$$

- Get new p_{EL}!!

$$= EU_P(R) = p_{EL} \cdot 0 + (1-p_{EL}) \cdot \frac{2^{1-r}}{1-r} = \frac{2^{1-r}(1-p_{EL})}{1-r}$$

Risk-Averse Logit-QRE Model

- Risk-Averse Nash Equilibrium with CRRA:
  - Constant Relative Risk Aversion (r = RRA):

$$U(x) = \frac{x^{1-r}}{1-r},\ x \geq 0$$

$$= -\frac{(-x)^{1-r}}{1-r},\ x < 0$$

Pursuer \ Evader		L (p_{EL})	R ($1-p_{EL}$)
p_{PL}	L	1, -1	0, 0
$1-p_{PL}$	R	0, 0	2, -2

$$EU_E(L) = p_{PL} \cdot \frac{-1}{1-r} + (1-p_{PL}) \cdot 0 = -\frac{p_{PL}}{1-r}$$

Get new p_{PL}!

$$= EU_E(R) = p_{PL} \cdot 0 + (1-p_{PL}) \cdot \frac{-2^{1-r}}{1-r} = -\frac{2^{1-r}(1-p_{PL})}{1-r}$$

Risk-Averse Logit-QRE Model

- Nash = Consistent Beliefs + Best Response
- QRE = Consistent Beliefs + Better Response
- Add stochastic term (noise) to expected payoffs

1. $EU_P^*(L) = \frac{p_{EL}}{1-r} + \epsilon_{PL}$
2. $EU_P^*(R) = \frac{2^{1-r}(1-p_{EL})}{1-r} + \epsilon_{PR}$
3. $EU_E^*(L) = -\frac{p_{PL}}{1-r} + \epsilon_{EL}$
4. $EU_E^*(R) = -\frac{2^{1-r}(1-p_{PL})}{1-r} + \epsilon_{ER}$

Evader

Pursuer		p_{EL} L	$1-p_{EL}$ R
p_{PL}	L	1, -1	0, 0
$1-p_{PL}$	R	0, 0	2, -2

Estimating Risk-Averse Logit-QRE with STATA

- Let errors be type-I Extreme Value distribution (μ)

$$\epsilon_i \overset{\text{iid}}{\sim} F(\epsilon_i;\mu) = \exp\left(-\exp\left(-\frac{\epsilon_i}{\mu}\right)\right)$$

- Payoff Response Function (with Noise Parameter μ)

$$p_{PL} = \frac{\exp\left[\frac{p_{EL}}{(1-r)\mu}\right]}{\exp\left[\frac{p_{EL}}{(1-r)\mu}\right] + \exp\left[\frac{2^{1-r}(1-p_{EL})}{(1-r)\mu}\right]}$$

  - $\mu = 0$: No Noise
  - $\mu = \infty$: All Noise

- Use MATA/ML to solve for μ and r

$$p_{EL} = \frac{\exp\left[-\frac{p_{PL}}{(1-r)\mu}\right]}{\exp\left[-\frac{p_{PL}}{(1-r)\mu}\right] + \exp\left[-\frac{2^{1-r}(1-p_{PL})}{(1-r)\mu}\right]}$$

Estimating Risk-Avers

- STATA Code:

- Choose p to
- Minimize this

```
* START MATA FROM WITHIN STATA

mata:

// SET STARTING VALUES FOR THE TWO PROBABILITIES

start=(0.5,0.5)
```

```
// CREATE PROGRAM ("vector_min") FOR EVALUATING 2x1 VECTOR (ss)
// WHOSE ELEMENTS ARE TO BE MINIMISED

void vector_min(todo, p, ss, S, H)
        {
 external X, mu, r
                PP    = p[1]
                PE    = p[2]

 EU_PL=PE/(mu*(1-r))
 EU_PR=(2^(1-r))*(1-PE)/(mu*(1-r))

 EU_EL=(-PP)/(mu*(1-r))
 EU_ER=-(2^(1-r))*(1-PP)/(mu*(1-r))

 s1=(PP-exp(EU_PL)/(exp(EU_PL)+exp(EU_PR)))^2
 s2=(PE-exp(EU_EL)/(exp(EU_EL)+exp(EU_ER)))^2

                ss= s1\s2

        }
```

Estimating Risk-A

- STATA Code:
- (Second-half of MATA)

```
// BEGIN DEFINITION OF OPTIMISATION PROBLEM,
// RETURNING S, A PROBLEM-DESCRIPTION HANDLE CONTAINING DEFAULT VALUES

 S = optimize_init()

// MODIFY DEFAULTS

optimize_init_evaluator(S, &vector_min())
 optimize_init_evaluatortype(S, "v0")
optimize_init_params(S, start)
 optimize_init_which(S,  "min" )
 optimize_init_tracelevel(S,"none")
 optimize_init_conv_ptol(S, 1e-16)
 optimize_init_conv_vtol(S, 1e-16)

// RETURN TO STATA

end
clear

* GENERATE DATA SET

set obs 1000
gen int y_P = _n<501
gen int y_E = _n<668
```

- Simulate 1000:
  - $n_{PL} = 500$
  - $n_{EL} = 667$

Estimating Risk-A

- STATA Code:

- 2 Initial vectors:
  - MATA: start
  - ML: sstart
    (see next slide)

```
* SET STARTING VALUES FOR COMPUTATION OF THE TWO PROBABILITIES

mat start=(0.67,0.67)

prog drop _all

* LOG-LIKELIHOOD EVALUATION PROGRAM (qre_risk) STARTS HERE:

program define qre_risk
 args lnf mu r
 tempvar pp
 tempname p1 p2 mmu rr

scalar mmu=`mu'
scalar rr=`r'

* COPY STARTING PROBABILITIES AND TWO PARAMETERS (mu and r) INTO MATA

mata: start=st_matrix("start")
mata: mu=st_numscalar("mmu")
mata: r=st_numscalar("rr")

* PERFORM OPTIMIZATION; STORE SOLUTION IN 2x1 VECTOR p:

mata: p = optimize(S)
```

Calling MATA to solve $p_{PL}(\mu)$, $p_{EL}(\mu)$

Estimating Ris

- STATA Code:

```
* EXTRACT ELEMENTS OF p

mata: st_numscalar("p1",p[1])
mata:  st_numscalar("p2", p[2])

* GENERATE JOINT PROBABILITY OF PURSUER'S AND EVADER'S DECISIONS:

 quietly gen double `pp'=((p1*y_P)+(1-p1)*(1-y_P))*((p2*y_E)+(1-p2)*(1-y_E))

* GENERATE LOG-LIKELIHOOD

 quietly replace `lnf'=ln(`pp')
 end

* SET STARTING VALUES FOR 2 PARAMETERS mu AND r

mat sstart=(.1,0.3)

* RUN ML

ml model lf qre_risk () ()
ml init sstart, copy
ml maximize
```

Evaluates $\log L$ for a given μ

Pick μ yielding largest $\log L$

Estimating R

- STATA Results:
- Risk-averse adds little:
  - $\mu = 0.72$ same
  - $r =$ -0.002 close to risk neutral

```
. ml maximize

initial:       log likelihood = -1354.7992
rescale:       log likelihood = -1353.2851
rescale eq:    log likelihood =  -1329.691
Iteration 0:   log likelihood =  -1329.691
Iteration 1:   log likelihood = -1329.4408
Iteration 2:   log likelihood = -1329.4301
Iteration 3:   log likelihood =   -1329.43

                                                Number of obs     =      1,000
                                                Wald chi2(0)      =          .
Log likelihood =  -1329.43                      Prob > chi2       =          .

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
eq1          |
       _cons |   .7203918   .1047659     6.88   0.000     .5150544    .9257293
-------------+----------------------------------------------------------------
eq2          |
       _cons |  -.0021648   .1382919    -0.02   0.988    -.2732119    .2688824
------------------------------------------------------------------------------
```

$\mu = 0.72 / r =$ -0.002 achieves maximum $\log L =$ -1329.43

Estimating R

- STATA C
- Results:

```
                                                Number of obs     =      1,000
                                                Wald chi2(0)      =          .
Log likelihood = -1366.1588
------------------------------------------------------------------------------
             |      Coef.   Std. Err.
-------------+----------------------------------------------------------------
eq1        μ |
       _cons |   1.054045   .2420504       4
------------------------------------------------------------------------------
eq2        r |
       _cons |   .4150363   .1321526     3.14   0.002      .156022    .6740506
------------------------------------------------------------------------------
```

$\mu = 1.05$ / $r = 0.415$ achieves maximum $\log L =$ -1366.1588

$r = 0.415$ far from Risk Neutrality!

- Instead Try:
  - $n_{PL} = 500$
  - $n_{EL} = 600$
    - Not 667

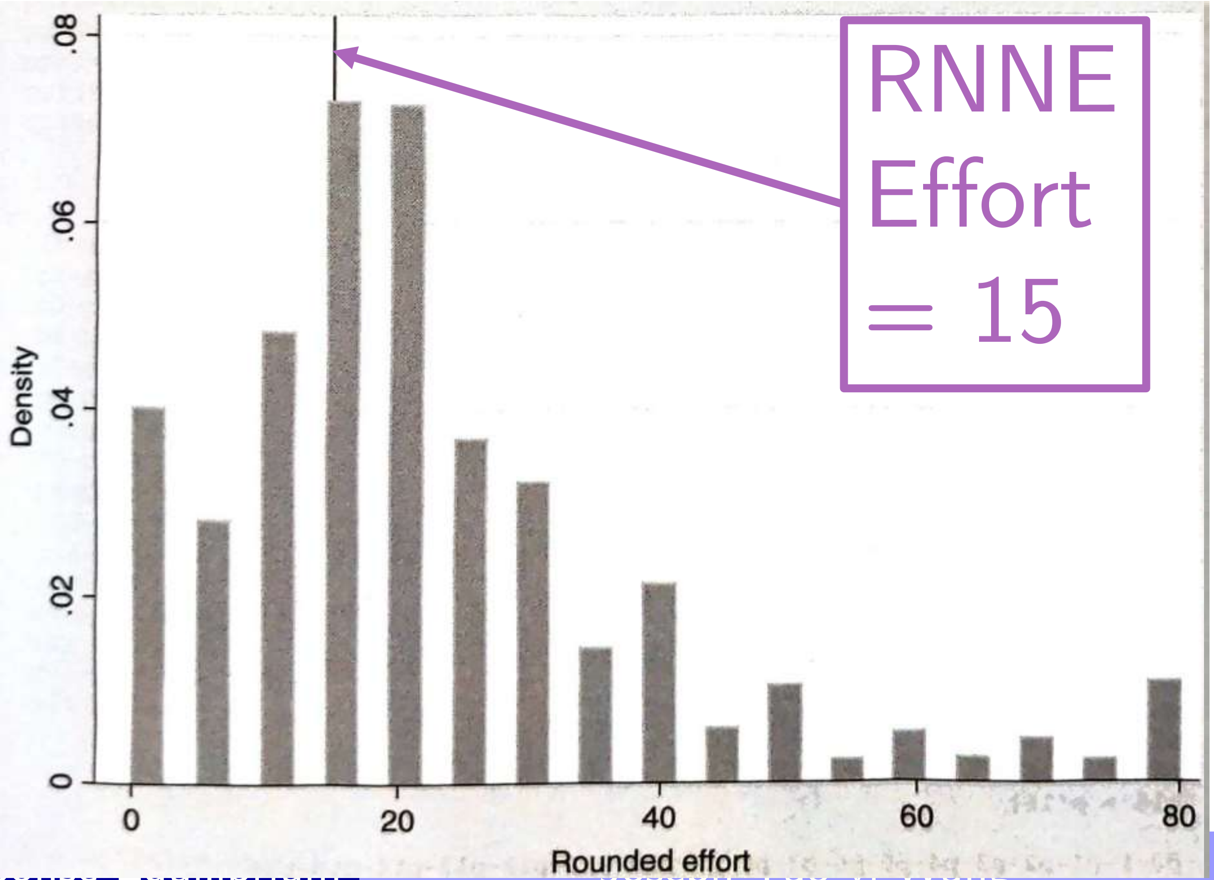
```
 optimize_i
 optimize_i

// RETURN TO STATA

end
clear

* GENERATE DATA SET

set obs 1000
gen int y_P = _n<501
gen int y_E = _n<601
```

$\mu = 1.05 > 1.39$ (risk neutral μ)

Noise partly due to risk aversion

Logit-QRE on Tullock Contest Data

- Tullock Contest Data of Chowdhury et al. (2014)
  - Chapter 4 of Moffatt (2016) analyzes `chowdhury.dta`
  - Can QRE explain overbidding beyond RNNE? (Figure 4.4)
- But effort = 0, 1, 2, ..., 80!
  - Hard to compute QRE...
- Data cluster at 0, 5, 10,...
  - Can round up to nearest 5!
  - `gen y=5*round(bid/5,1)`



Logit-QRE on Tullock Contest Data

- Believe others choose $(y^0, y^1, y^2, \cdots, y^{16}) = (0, 5, 10, \cdots, 80)$
- with probability $(p^0, p^1, p^2, \cdots, p^{16})$

- Expect Total Effort by opponents to be $(n-1)\sum_{j=0}^{16} p^j y^j$
- Expected Payoff for choosing effort y is

$$EV_i(y; p^0, \cdots, p^{16}) = \frac{80y}{y + (n-1)\sum_{j=0}^{16} p^j y^j}, \; y = y^0, \cdots, y^{16}$$

- Quantal Response (QR):

$$p_i(y^j; p^0, \cdots, p^{16}; \mu) = \frac{\exp\left[EV_i(y^j)/\mu\right]}{\sum_{k=0}^{16} \exp\left[EV_i(y^k)/\mu\right]}$$

Logit-QRE on Tullock Contest Data

- Quantal Response (QR):

$$p_i(y^j; p^0, \cdots, p^{16}; \mu) = \frac{\exp\left[EV_i(y^j)/\mu\right]}{\sum_{k=0}^{16} \exp\left[EV_i(y^k)/\mu\right]}$$

- QRE (for given μ) is $(\tilde{p}^0(\mu), \cdots, \tilde{p}^{16}(\mu))$ such that (j=0-16)

$$p_i(y^j; \tilde{p}^0, \cdots, \tilde{p}^{16}; \mu) = \tilde{p}^j(\mu)$$

  - If efforts $y_1, \ldots, y_N$ is iid, $\log L(\mu) = \sum_{i=1}^{N} \sum_{j=0}^{16} I(y_i = y^j) \ln\left[\tilde{p}^j(\mu)\right]$

- MATA solves fixed point $(\tilde{p}^0(\mu), \cdots, \tilde{p}^{16}(\mu))$ for given μ
  - ML finds likelihood-maximizing μ

Logit-QRE

- STATA Code:

```
void mysolver(todo, p, fff, S, H)
    {
 external mu
p1 = p[1]
p2 = p[2]
p3 = p[3]
p4 = p[4]
p5 = p[5]
p6 = p[6]
p7 = p[7]
p8 = p[8]
p9 = p[9]
p10 = p[10]
p11 = p[11]
p12 = p[12]
p13 = p[13]
p14 = p[14]
p15 = p[15]
p16 = p[16]


p0=1-p1-p2-p3-p4-p5-p6-p7-p8-p9-p10-p11-p12-p13-p14-p15-p16

ee=5*(0*p0+1*p1+2*p2+3*p3+4*p4+5*p5+6*p6+7*p7+8*p8+9*p9+10*p10  ///
 +11*p11+12*p12+13*p13+14*p14+15*p15+16*p16)
```

```
*  Maximum likelihood estimation of "mu" in QRE model for contest data.
*  MATA Optimize routine is used to find p0-p16 for given mu.
*  STATA ML routine maximises the log-likelihood over mu

clear
clear mata
set more off

mata:

start=(0.08,0.08,0.08,0.08,0.04,0.04,0.04,0.04,0.04,0.04, ///
0.04,0.04,0.04,0.04,0.04,0.04)
```

Logit-QI

- STAT
  Code:

```
zz0=exp(0*(80/(0+3*ee)-1)/mu)
 zz1=exp(5*(80/(5+3*ee)-1)/mu)
  zz2=exp(10*(80/(10+3*ee)-1)/mu)
   zz3=exp(15*(80/(15+3*ee)-1)/mu)
 zz4=exp(20*(80/(20+3*ee)-1)/mu)
  zz5=exp(25*(80/(25+3*ee)-1)/mu)
zz6=exp(30*(80/(30+3*ee)-1)/mu)
 zz7=exp(35*(80/(35+3*ee)-1)/mu)
  zz8=exp(40*(80/(40+3*ee)-1)/mu)
   zz9=exp(45*(80/(45+3*ee)-1)/mu)
 zz10=exp(50*(80/(50+3*ee)-1)/mu)

zz11=exp(55*(80/(55+3*ee)-1)/mu)
  zz12=exp(60*(80/(60+3*ee)-1)/mu)
   zz13=exp(65*(80/(65+3*ee)-1)/mu)
 zz14=exp(70*(80/(70+3*ee)-1)/mu)
  zz15=exp(75*(80/(75+3*ee)-1)/mu)
zz16=exp(80*(80/(80+3*ee)-1)/mu)


zz=zz0+zz1+zz2+zz3+zz4+zz5+zz6+zz7+zz8+zz9+zz10+zz11 ///
+ zz12+zz13+zz14+zz15+zz16
```

```
pp0=zz0/zz
pp1=zz1/zz
pp2=zz2/zz
pp3=zz3/zz
pp4=zz4/zz
pp5=zz5/zz
pp6=zz6/zz
pp7=zz7/zz
pp8=zz8/zz
pp9=zz9/zz
pp10=zz10/zz

pp11=zz11/zz
pp12=zz12/zz
pp13=zz13/zz
pp14=zz14/zz
pp15=zz15/zz
pp16=zz16/zz
```

```
fff=(p1-pp1)^2 \
(p2-pp2)^2 \
(p3-pp3)^2 \
(p4-pp4)^2 \
(p5-pp5)^2 \
(p6-pp6)^2 \
(p7-pp7)^2 \
(p8-pp8)^2 \
(p9-pp9)^2 \
(p10-pp10)^2 \
 (p11-pp11)^2 \
(p12-pp12)^2 \
(p13-pp13)^2 \
(p14-pp14)^2 \
(p15-pp15)^2 \
(p16-pp16)^2


     }
```

Logit-QRE

- STATA Code:

```
 S = optimize_init()

optimize_init_evaluator(S, &mysolver())
 optimize_init_evaluatortype(S, "v0")
optimize_init_params(S, start)
 optimize_init_which(S,  "min" )
 optimize_init_tracelevel(S,"none")
 optimize_init_conv_ptol(S, 1e-16)
 optimize_init_conv_vtol(S, 1e-16)

end

clear

scalar mu=10.0

mat start=(1.101704)

prog drop _all

 mata: start=st_matrix("start")

mata: mu=st_numscalar("mu")
```

```
mata: p = optimize(S)

mata: start=p

mata: st_numscalar("p1",p[1])
mata: st_numscalar("p2",p[2])
mata: st_numscalar("p3",p[3])
mata: st_numscalar("p4",p[4])
mata: st_numscalar("p5",p[5])
mata: st_numscalar("p6",p[6])
mata: st_numscalar("p7",p[7])
mata: st_numscalar("p8",p[8])
mata: st_numscalar("p9",p[9])
mata: st_numscalar("p10",p[10])

mata: st_numscalar("p11",p[11])
mata: st_numscalar("p12",p[12])
mata: st_numscalar("p13",p[13])
mata: st_numscalar("p14",p[14])
mata: st_numscalar("p15",p[15])
mata: st_numscalar("p16",p[16])
```

```
scalar p0=1-p1-p2-p3-p4-p5-p6-p7-p8-p9-p10-p11-p12-p13-p14-p15-p16

clear

mat start=(0.08,0.08,0.08,0.08,0.04,0.04,0.04, ///
0.04,0.04,0.04,0.04,0.04,0.04,0.04,0.04,0.04)

prog drop _all

program define qre
 args lnf mu
 tempvar pp
 tempname p0 p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14 p15 p16 mmu

scalar mmu=`mu'

mata: start=st_matrix("start")

mata: mu=st_numscalar("mmu")
mata: X=st_numscalar("X")

mata: p = optimize(S)

mata: start=p
```

```
mata: st_numscalar("p1",p[1])
mata:  st_numscalar("p2", p[2])
mata: st_numscalar("p3",p[3])
mata:  st_numscalar("p4", p[4])
mata: st_numscalar("p5",p[5])
mata:  st_numscalar("p6", p[6])
mata:  st_numscalar("p7", p[7])
mata: st_numscalar("p8",p[8])
mata:  st_numscalar("p9", p[9])
mata:  st_numscalar("p10", p[10])
mata: st_numscalar("p11",p[11])
mata:  st_numscalar("p12", p[12])
mata: st_numscalar("p13",p[13])
```

```
mata:  st_numscalar("p14", p[14])
mata: st_numscalar("p15",p[15])
mata:  st_numscalar("p16", p[16])

scalar p0=1-p1-p2-p3-p4-p5-p6-p7-p8-p9-p10-p11-p12-p13-p14-p15-p16

forvalues k=1(1)16 {
scalar p`k'=max(p`k',1.0e-12)
}

scalar p0=1-p1-p2-p3-p4-p5-p6-p7-p8-p9-p10-p11-p12-p13-p14-p15-p16
```

Logit-QRE

- STATA Code:

```
quietly gen double ‘pp’=y0*p0+y1*p1+y2*p2+y3*p3+y4*p4+y5*p5+y6*p6 ///
 +y7*p7+y8*p8+y9*p9+y10*p10+y11*p11+y12*p12+y13*p13+y14*p14+y15*p15+y16*p16

quietly replace ‘lnf’=ln(‘pp’)
end

use chowdhury, clear

keep if t>15

* ROUND BID TO NEAREST 5

gen y=5*round(bid/5,1)

gen y0=y==0
gen y1=y==5
gen y2=y==10
gen y3=y==15
gen y4=y==20
gen y5=y==25
gen y6=y==30
gen y7=y==35
gen y8=y==40
gen y9=y==45
gen y10=y==50
gen y11=y==55
gen y12=y==60
gen y13=y==65
gen y14=y==70
gen y15=y==75
gen y16=y==80

mat sstart=(10)

ml model lf qre ()
ml init sstart, copy
ml maximize
```

Logit-QRE on Tullock Contest Data

- STATA Results:
- μ = 10.42 with CI = [9.56, 11.27] with logL = -5433.8
- Chowdhury et al. (2014): Table 4
  - logL = -5424.5
  - But estimate $\lambda = 1/\mu$ instead of μ

Conclusion: QRE

- Two Tests for Repeated Observation of Game Data:
  - Binomial Test
  - Runs Test
- QRE
  - Better Response (instead of Best Response)
  - Computed via (Non-Linear) Minimization
  - Risk-Averse QRE using CRRA
  - Application to Contest Data

Acknowledgment

- This presentation is based on
  - Section 16.1-6 of the textbook on Experimetrics, an
- Extension of a mini-course taught by Peter G. Moffatt (UEA) at National Taiwan University in Spring 2019
  - We would like to thank 施帆倢, Samira Basaka Nanje and Roariker Anthony Swenson for their in-class presentations