Robust income distribution analysis

Philippe Van Kerm

CEPS/INSTEAD
Luxembourg

5th German Stata Users Group Meeting (RWI Essen, April 2, 2007)

[outline]

1. The problem of data contamination/extreme incomes
2. Robust estimation strategies
3. Stata Implementation of OBRE
4. A brief empirical illustration
5. Concluding remarks

The problem of data contamination/extreme incomes

- Income distribution analysis:
  1. summary measures of inequality (and other distributional features)
  2. dominance checks (stochastic dominance, Lorenz dominance)
- Both very sensitive to extreme incomes ('valid' outliers? contamination?)
  - unbounded influence function (Cowell & Victoria-Feser, *Econometrica* 1996, 2002)

The problem of data contamination/extreme incomes

- Income distribution analysis:
  1. summary measures of inequality (and other distributional features)
  2. dominance checks (stochastic dominance, Lorenz dominance)
- Both very sensitive to extreme incomes ('valid' outliers? contamination?)
  - unbounded influence function (Cowell & Victoria-Feser, *Econometrica* 1996, 2002)

The problem of data contamination/extreme incomes

PSELL-3 (equivalised) household income data (waves 1-3):

	2002			2003			2004		
	Top 10 incomes								
		37,260			16,925			41,830	
		34,242			15,280			32,569	
		28,292			15,132			18,341	
		...			...			...	
		15,407			10,464			11,095	
	Summary measures								
	Raw	Trim	Wins.	Raw	Trim	Wins.	Raw	Trim	Wins.
μ	2,689	2,635	2,666	2,674	2,631	2,667	2,734	2,685	2,715
Gini	0.272	0.259	0.266	0.262	0.252	0.259	0.262	0.250	0.257
$\frac{CV^2}{2}$	0.192	0.129	0.147	0.138	0.116	0.129	0.159	0.112	0.123

The problem of data contamination/extreme incomes

PSELL-3 (equivalised) household income data (waves 1-3):

	2002			2003			2004		
	Top 10 incomes								
		37,260			16,925			41,830	
		34,242			15,280			32,569	
		28,292			15,132			18,341	
		...			...			...	
		15,407			10,464			11,095	
	Summary measures								
	Raw	Trim	Wins.	Raw	Trim	Wins.	Raw	Trim	Wins.
μ	2,689	2,635	2,666	2,674	2,631	2,667	2,734	2,685	2,715
Gini	0.272	0.259	0.266	0.262	0.252	0.259	0.262	0.250	0.257
$\frac{CV^2}{2}$	0.192	0.129	0.147	0.138	0.116	0.129	0.159	0.112	0.123

The problem of data contamination/extreme incomes

PSELL-3 (equivalised) household income data (waves 1-3):

	2002			2003			2004		
	Top 10 incomes								
		37,260			16,925			41,830	
		34,242			15,280			32,569	
		28,292			15,132			18,341	
		...			...			...	
		15,407			10,464			11,095	
	Summary measures								
	Raw	Trim	Wins.	Raw	Trim	Wins.	Raw	Trim	Wins.
μ	2,689	2,635	2,666	2,674	2,631	2,667	2,734	2,685	2,715
Gini	0.272	0.259	0.266	0.262	0.252	0.259	0.262	0.250	0.257
$\frac{CV^2}{2}$	0.192	0.129	0.147	0.138	0.116	0.129	0.159	0.112	0.123

The problem of data contamination/extreme incomes

PSELL-3 (equivalised) household income data (waves 1-3):

	2002			2003			2004		
	Top 10 incomes								
		37,260			16,925			41,830	
		34,242			15,280			32,569	
		28,292			15,132			18,341	
		...			...			...	
		15,407			10,464			11,095	
	Summary measures								
	Raw	Trim	Wins.	Raw	Trim	Wins.	Raw	Trim	Wins.
μ	2,689	2,635	2,666	2,674	2,631	2,667	2,734	2,685	2,715
Gini	0.272	0.259	0.266	0.262	0.252	0.259	0.262	0.250	0.257
$\frac{cv^2}{2}$	0.192	0.129	0.147	0.138	0.116	0.129	0.159	0.112	0.123

[outline]

Remedial actions

1. Remove extremely high incomes, or impose a top code
   - Easy, but not efficient and dependence to trimming fractions
2. Use functional form assumptions:
   - model upper tail of distribution parametrically (e.g. Pareto distribution)[1]
   - model the full distribution parametrically (e.g. log-Normal, Gamma, Singh-Maddala)
   - But... classical ML estimators are themselves non-robust to extreme incomes!

[1] See, e.g., Cowell & Victoria-Feser, *Journal of Economic Inequality* 2007; Cowell & Flachaire, *Journal of Econometrics* 2007

Remedial actions

1. Remove extremely high incomes, or impose a top code
   - Easy, but not efficient and dependence to trimming fractions
2. Use functional form assumptions:
   - model upper tail of distribution parametrically (e.g. Pareto distribution)[1]
   - model the full distribution parametrically (e.g. log-Normal, Gamma, Singh-Maddala)
   - But... classical ML estimators are themselves non-robust to extreme incomes!

[1] See, e.g., Cowell & Victoria-Feser, *Journal of Economic Inequality* 2007; Cowell & Flachaire, *Journal of Econometrics* 2007

Remedial actions

1. Remove extremely high incomes, or impose a top code
   - Easy, but not efficient and dependence to trimming fractions
2. Use functional form assumptions:
   - model upper tail of distribution parametrically (e.g. Pareto distribution)[1]
   - model the full distribution parametrically (e.g. log-Normal, Gamma, Singh-Maddala)
   - But... classical ML estimators are themselves non-robust to extreme incomes!

[1] See, e.g., Cowell & Victoria-Feser, *Journal of Economic Inequality* 2007; Cowell & Flachaire, *Journal of Econometrics* 2007

Remedial actions

1. Remove extremely high incomes, or impose a top code
   - Easy, but not efficient and dependence to trimming fractions
2. Use functional form assumptions:
   - model upper tail of distribution parametrically (e.g. Pareto distribution)[1]
   - model the full distribution parametrically (e.g. log-Normal, Gamma, Singh-Maddala)
   - But... classical ML estimators are themselves non-robust to extreme incomes!

[1] See, e.g., Cowell & Victoria-Feser, *Journal of Economic Inequality* 2007; Cowell & Flachaire, *Journal of Econometrics* 2007

Remedial actions

1. Remove extremely high incomes, or impose a top code
   - Easy, but not efficient and dependence to trimming fractions
2. Use functional form assumptions:
   - model upper tail of distribution parametrically (e.g. Pareto distribution)[1]
   - model the full distribution parametrically (e.g. log-Normal, Gamma, Singh-Maddala)
   - But... classical ML estimators are themselves non-robust to extreme incomes!

[1] See, e.g., Cowell & Victoria-Feser, *Journal of Economic Inequality* 2007; Cowell & Flachaire, *Journal of Econometrics* 2007

Remedial actions

1. Remove extremely high incomes, or impose a top code
   - Easy, but not efficient and dependence to trimming fractions
2. Use functional form assumptions:
   - model upper tail of distribution parametrically (e.g. Pareto distribution)[1]
   - model the full distribution parametrically (e.g. log-Normal, Gamma, Singh-Maddala)
   - But... classical ML estimators are themselves non-robust to extreme incomes!

[1] See, e.g., Cowell & Victoria-Feser, *Journal of Economic Inequality* 2007; Cowell & Flachaire, *Journal of Econometrics* 2007

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

- OBRE is an M-estimator: θ solution to $\sum_{i=1}^{N} \psi(x_i, \theta) = 0$
  - (For ML: $\psi(x_i, \theta^{ML}) = s(x_i, \theta^{ML})$ is the score function)
- OBRE estimator is a special M-estimator with

$$\psi(x_i, \theta^{OB}) = \left(s(x_i, \theta^{OB}) - a(\theta^{OB})\right) W_c(x_i; \theta^{OB})$$

where

$$W_c(x_i; \theta^{OB}) = \min\left(1; \frac{c}{G(s(x_i, \theta^{OB}), a(\theta^{OB}), A(\theta^{OB}))}\right)$$

- $W_c(x; \theta^{OB})$ imposes a bound on influence function by downweighting extreme values (values deviating from model)
- c is a ‘robustness’ parameter to be determined (efficiency-robustness trade-off)

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

- OBRE is an M-estimator: θ solution to $\sum_{i=1}^{N} \psi(x_i, \theta) = 0$
  - (For ML: $\psi(x_i, \theta^{ML}) = s(x_i, \theta^{ML})$ is the score function)
- OBRE estimator is a special M-estimator with

$$\psi(x_i, \theta^{OB}) = (s(x_i, \theta^{OB}) - a(\theta^{OB}))W_c(x_i; \theta^{OB})$$

  where

$$W_c(x_i; \theta^{OB}) = \min\left(1; \frac{c}{G(s(x_i, \theta^{OB}), a(\theta^{OB}), A(\theta^{OB}))}\right)$$

- $W_c(x; \theta^{OB})$ imposes a bound on influence function by downweighting extreme values (values deviating from model)
- c is a 'robustness' parameter to be determined (efficiency-robustness trade-off)

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

- OBRE is an M-estimator: θ solution to $\sum_{i=1}^{N} \psi(x_i, \theta) = 0$
  - (For ML: $\psi(x_i, \theta^{ML}) = s(x_i, \theta^{ML})$ is the score function)
- OBRE estimator is a special M-estimator with

$$\psi(x_i, \theta^{OB}) = (s(x_i, \theta^{OB}) - a(\theta^{OB}))W_c(x_i; \theta^{OB})$$

where

$$W_c(x_i; \theta^{OB}) = \min\left(1; \frac{c}{G(s(x_i, \theta^{OB}), a(\theta^{OB}), A(\theta^{OB}))}\right)$$

- $W_c(x; \theta^{OB})$ imposes a bound on influence function by downweighting extreme values (values deviating from model)
- c is a ‘robustness’ parameter to be determined (efficiency-robustness trade-off)

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

- OBRE is an M-estimator: θ solution to $\sum_{i=1}^{N} \psi(x_i, \theta) = 0$
  - (For ML: $\psi(x_i, \theta^{ML}) = s(x_i, \theta^{ML})$ is the score function)
- OBRE estimator is a special M-estimator with

$$\psi(x_i, \theta^{OB}) = (s(x_i, \theta^{OB}) - a(\theta^{OB}))W_c(x_i; \theta^{OB})$$

  where

$$W_c(x_i; \theta^{OB}) = \min\left(1; \frac{c}{G(s(x_i, \theta^{OB}), a(\theta^{OB}), A(\theta^{OB}))}\right)$$

- $W_c(x; \theta^{OB})$ imposes a bound on influence function by downweighting extreme values (values deviating from model)
- c is a 'robustness' parameter to be determined (efficiency-robustness trade-off)

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

- OBRE is an M-estimator: θ solution to $\sum_{i=1}^{N} \psi(x_i, \theta) = 0$
  - (For ML: $\psi(x_i, \theta^{ML}) = s(x_i, \theta^{ML})$ is the score function)
- OBRE estimator is a special M-estimator with

$$\psi(x_i, \theta^{OB}) = (s(x_i, \theta^{OB}) - a(\theta^{OB}))W_c(x_i; \theta^{OB})$$

  where

$$W_c(x_i; \theta^{OB}) = \min\left(1; \frac{c}{G(s(x_i, \theta^{OB}), a(\theta^{OB}), A(\theta^{OB}))}\right)$$

- $W_c(x; \theta^{OB})$ imposes a bound on influence function by downweighting extreme values (values deviating from model)
- c is a 'robustness' parameter to be determined (efficiency-robustness trade-off)

Optimal B-Robust Estimators (OBRE) (ctd.)

A robust alternative to classical ML

- $a(\theta^{OB})$ and $A(\theta^{OB})$ are such that

$$\begin{aligned} E(\psi(x,\theta^{OB})\psi(x,\theta^{OB})') &= (A(\theta^{OB})A(\theta^{OB})')^{-1} \\ E(\psi(x,\theta^{OB})) &= 0 \end{aligned}$$

The resulting estimator is the optimal (minimum variance) M-estimator with bounded influence function[2]

- If $c \rightarrow \infty$ then $\theta^{OB} = \theta^{ML}$

[2] For a thorough treatment, see Hampel et al. (1986), *Robust Statistics: The approach based on influence functions.*

Optimal B-Robust Estimators (OBRE) (ctd.)

A robust alternative to classical ML

- $a(\theta^{OB})$ and $A(\theta^{OB})$ are such that

$$\begin{aligned} E(\psi(x,\theta^{OB})\psi(x,\theta^{OB})') &= (A(\theta^{OB})A(\theta^{OB})')^{-1} \\ E(\psi(x,\theta^{OB})) &= 0 \end{aligned}$$

  The resulting estimator is the optimal (minimum variance) M-estimator with bounded influence function[2]

- If $c \rightarrow \infty$ then $\theta^{OB} = \theta^{ML}$

[2] For a thorough treatment, see Hampel et al. (1986), *Robust Statistics: The approach based on influence functions*.

Optimal B-Robust Estimators (OBRE) (ctd.)

A robust alternative to classical ML

- $a(\theta^{OB})$ and $A(\theta^{OB})$ are such that

$$\begin{aligned} E(\psi(x,\theta^{OB})\psi(x,\theta^{OB})') &= (A(\theta^{OB})A(\theta^{OB})')^{-1} \\ E(\psi(x,\theta^{OB})) &= 0 \end{aligned}$$

The resulting estimator is the optimal (minimum variance) M-estimator with bounded influence function[2]

- If $c \to \infty$ then $\theta^{OB} = \theta^{ML}$

[2] For a thorough treatment, see Hampel et al. (1986), *Robust Statistics: The approach based on influence functions*.

[outline]

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I follow Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm involving
  1. matrix operations
  2. numerical integration

  $\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I follow Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm involving
  1. matrix operations
  2. numerical integration

  $\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I follow Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm involving
  1. matrix operations
  2. numerical integration

  $\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I follow Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm involving
  1. matrix operations
  2. numerical integration

  $\Longrightarrow$ Mata!

Implementation (ctd.)

- Implementation is relatively easy with Mata (but familiarity with matrix algebra can help!)
- Builds on suite of commands by Stephen Jenkins to fit functional forms to unit record data by ML[3]
  - just replace ML engine by home-brewed OBRE engine (call a Mata function, rather than `ml model`)
- I implemented Pareto Type I distribution and 3-parameters Singh-Maddala distribution[4]
- Compatible with Nick Cox's diagnostic commands `psm` and `qsm`

[3] `ssc describe smfit`

[4] `paretofit` to fit Pareto by ML will soon be added to Jenkins' collection.

Implementation (ctd.)

- Implementation is relatively easy with Mata (but familiarity with matrix algebra can help!)
- Builds on suite of commands by Stephen Jenkins to fit functional forms to unit record data by ML[3]
    - just replace ML engine by home-brewed OBRE engine (call a Mata function, rather than `ml model`)
- I implemented Pareto Type I distribution and 3-parameters Singh-Maddala distribution[4]
- Compatible with Nick Cox's diagnostic commands `psm` and `qsm`

[3] `ssc describe smfit`

[4] `paretofit` to fit Pareto by ML will soon be added to Jenkins' collection.

Implementation (ctd.)

- Implementation is relatively easy with Mata (but familiarity with matrix algebra can help!)
- Builds on suite of commands by Stephen Jenkins to fit functional forms to unit record data by ML[3]
  - just replace ML engine by home-brewed OBRE engine (call a Mata function, rather than `ml model`)
- I implemented Pareto Type I distribution and 3-parameters Singh-Maddala distribution[4]
- Compatible with Nick Cox's diagnostic commands `psm` and `qsm`

[3] `ssc describe smfit`

[4] `paretofit` to fit Pareto by ML will soon be added to Jenkins' collection.

Implementation (ctd.)

- Implementation is relatively easy with Mata (but familiarity with matrix algebra can help!)
- Builds on suite of commands by Stephen Jenkins to fit functional forms to unit record data by ML[3]
    - just replace ML engine by home-brewed OBRE engine (call a Mata function, rather than `ml model`)
- I implemented Pareto Type I distribution and 3-parameters Singh-Maddala distribution[4]
- Compatible with Nick Cox's diagnostic commands `psm` and `qsm`

[3] `ssc describe smfit`
[4] `paretofit` to fit Pareto by ML will soon be added to Jenkins' collection.

Implementation (ctd.)

- Implementation is relatively easy with Mata (but familiarity with matrix algebra can help!)
- Builds on suite of commands by Stephen Jenkins to fit functional forms to unit record data by ML[3]
  - just replace ML engine by home-brewed OBRE engine (call a Mata function, rather than `ml model`)
- I implemented Pareto Type I distribution and 3-parameters Singh-Maddala distribution[4]
- Compatible with Nick Cox's diagnostic commands `psm` and `qsm`

[3] `ssc describe smfit`
[4] `paretofit` to fit Pareto by ML will soon be added to Jenkins' collection.

Practical programming issues

- Precision of numerical integration functions revealed very important
- Difficulty to set multiple tolerance and precision parameters – trade-off between speed and accuracy (still subject to changes...)
- As in ML estimation, using re-parameterization $\tilde{\theta} = \ln(\theta)$ can help convergence (in all models considered, $\theta > 0$)

Practical programming issues

- Precision of numerical integration functions revealed very important
- Difficulty to set multiple tolerance and precision parameters – trade-off between speed and accuracy (still subject to changes...)
- As in ML estimation, using re-parameterization $\tilde{\theta} = \ln(\theta)$ can help convergence (in all models considered, $\theta > 0$)

Practical programming issues

- Precision of numerical integration functions revealed very important
- Difficulty to set multiple tolerance and precision parameters – trade-off between speed and accuracy (still subject to changes...)
- As in ML estimation, using re-parameterization $\tilde{\theta} = \ln(\theta)$ can help convergence (in all models considered, $\theta > 0$)

[outline]

Empirical illustration

- Data from panel survey PSELL-3 (Panel ‘Living in Luxembourg’), 2003–2005
- Representative of Luxembourg residents
- Single-adult-equivalent real household income (incomes of 2002-2004)

ML vs. OBRE parameter estimates

Pareto Type I parameters

		ML	OBRE			
			$c = 200$	$c = 5$	$c = 3$	$c = 2$
Pareto Type I	2002	3.635	3.635	3.633	3.720	3.926
(upper 5%)	2003	4.075	4.075	4.060	4.007	3.911
	2004	4.306	4.306	4.383	4.425	4.498

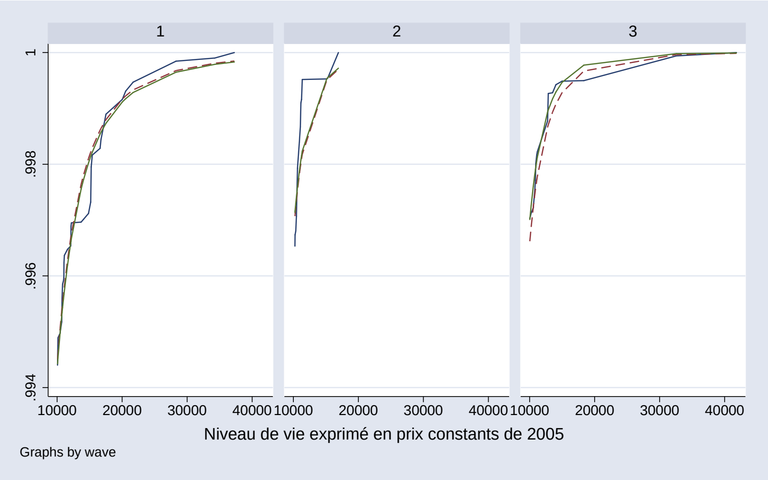
ML vs. OBRE parameter estimates

Pareto Type I parameters

		ML	OBRE			
			$c = 200$	$c = 5$	$c = 3$	$c = 2$
Pareto Type I	2002	3.635	3.635	3.633	3.720	3.926
(upper 5%)	2003	4.075	4.075	4.060	4.007	3.911
	2004	4.306	4.306	4.383	4.425	4.498

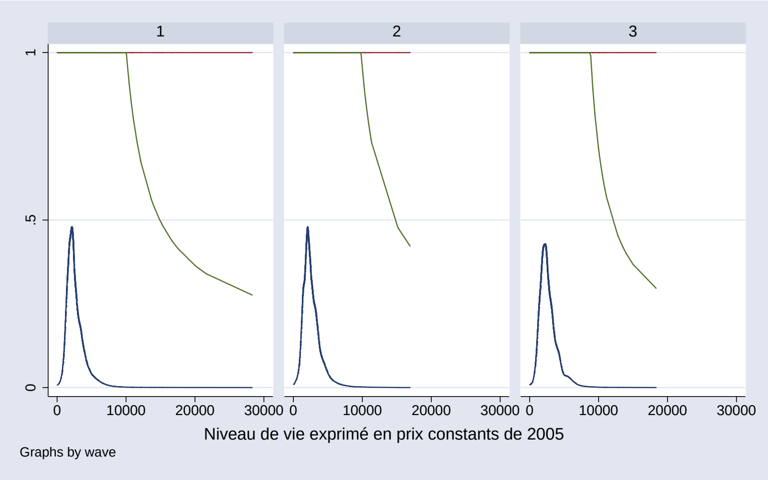
ML vs. OBRE parameter estimates

Empirical CDF and estimated Pareto Type I CDF



OBRE robustness weights

Pareto Type I distribution



ML vs. OBRE parameter estimates

Singh-Maddala parameters

		ML	OBRE			
			$c = 200$	$c = 10$	$c = 5$	$c = 4$
Singh-Maddala	2002	4.131	4.141	4.170	4.417	4.726
		2,159	2,159	2,146	2,022	1,912
		0.797	0.797	0.784	0.664	0.555
	2003	3.643	3.463	3.713	4.035	4.326
		2,477	2,477	2,428	2,214	2,060
		1.094	1.094	1.040	0.822	0.666
	2004	3.666	3.666	3.716	3.980	4.262
		2,529	2,529	2,496	2,278	2,124
		1.091	1.091	1.058	0.841	0.684

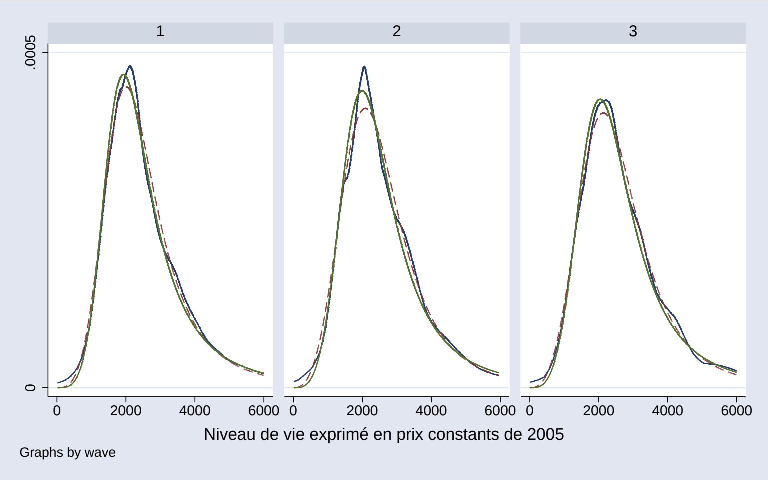
ML vs. OBRE parameter estimates

Singh-Maddala parameters

		ML	OBRE			
			$c = 200$	$c = 10$	$c = 5$	$c = 4$
Singh-Maddala	2002	4.131	4.141	4.170	4.417	4.726
		2,159	2,159	2,146	2,022	1,912
		0.797	0.797	0.784	0.664	0.555
	2003	3.643	3.463	3.713	4.035	4.326
		2,477	2,477	2,428	2,214	2,060
		1.094	1.094	1.040	0.822	0.666
	2004	3.666	3.666	3.716	3.980	4.262
		2,529	2,529	2,496	2,278	2,124
		1.091	1.091	1.058	0.841	0.684

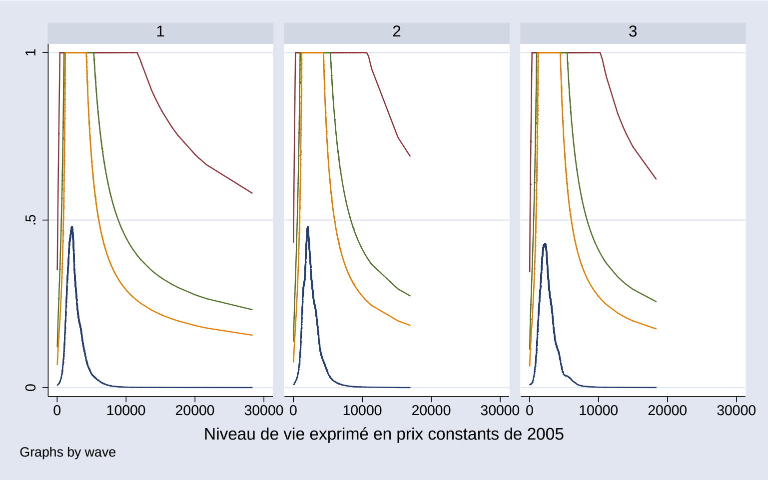
ML vs. OBRE parameter estimates

Non-parametric estimates and estimated Singh-Maddala PDFs



OBRE robustness weights

Singh-Maddala distribution



Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in income distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is a prototype (but looks ok). Minor developments still needed for
  - fixing precision and tolerance thresholds
  - allowing `svy:` prefix (?)
  - adding additional distributions (log-normal, gamma, Dagum) (?) – transplanting code to other distributions is easy

Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in income distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is a prototype (but looks ok). Minor developments still needed for
  - fixing precision and tolerance thresholds
  - allowing `svy:` prefix (?)
  - adding additional distributions (log-normal, gamma, Dagum) (?) – transplanting code to other distributions is easy

Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in income distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is a prototype (but looks ok). Minor developments still needed for
  - fixing precision and tolerance thresholds
  - allowing `svy:` prefix (?)
  - adding additional distributions (log-normal, gamma, Dagum) (?) – transplanting code to other distributions is easy

Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in income distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is a prototype (but looks ok). Minor developments still needed for
  - fixing precision and tolerance thresholds
  - allowing `svy:` prefix (?)
  - adding additional distributions (log-normal, gamma, Dagum) (?) – transplanting code to other distributions is easy

Concluding remarks (ctd.)

- Now entering more thorough testing phase:
  - Monte-Carlo simulations
  - Benchmarking against the software `IneQ` (by Cowell and Gomulka)
  - Further applications
- Then need to develop add-on software to help exploit these tools for deriving complete, robust inequality/poverty estimates

Concluding remarks (ctd.)

- Now entering more thorough testing phase:
  - Monte-Carlo simulations
  - Benchmarking against the software `IneQ` (by Cowell and Gomulka)
  - Further applications
- Then need to develop add-on software to help exploit these tools for deriving complete, robust inequality/poverty estimates

Concluding remarks (ctd.)

- Now entering more thorough testing phase:
  - Monte-Carlo simulations
  - Benchmarking against the software `IneQ` (by Cowell and Gomulka)
  - Further applications
- Then need to develop add-on software to help exploit these tools for deriving complete, robust inequality/poverty estimates