Extreme values and robust distribution analysis

Philippe Van Kerm

CEPS/INSTEAD, Luxembourg

ISER, University of Essex

13th UK Stata Users Group meeting
Cass Business School (London), September 10-11, 2007

[outline]

[outline]

Context

"Distribution analysis"

Analysis of data modelled as realizations from some random variable Y

- characterize Y w.r.t. 'location', 'spread'/'skewness', 'modality'
- focus on other particular features, e.g.
  - measures of inequality, poverty, polarization (income data)
  - expected loss, value-at-risk (financial data)
- stochastic dominance comparisons (ordering RV w.r.t. risk or inequality)
- fit parametric models for the RV (e.g., Gamma distribution, Pareto, etc.)

Context

"Distribution analysis"

Analysis of data modelled as realizations from some random variable Y

- characterize Y w.r.t. 'location', 'spread'/'skewness', 'modality'
- focus on other particular features, e.g.
  - measures of inequality, poverty, polarization (income data)
  - expected loss, value-at-risk (financial data)
- stochastic dominance comparisons (ordering RV w.r.t. risk or inequality)
- fit parametric models for the RV (e.g., Gamma distribution, Pareto, etc.)

The problem of data contamination and extreme values

The problem

Analysis beyond 'central tendency'/'location' estimation (very) sensitive to extreme data

- data contamination (e.g., 'decimal point' encoding error')?
- 'valid' outliers?

Consequences are potential bias and high sampling uncertainty (even with large samples).

$\Longrightarrow$ Many measures of interest have 'unbounded influence function'

The problem of data contamination and extreme values

The problem

Analysis beyond ‘central tendency’/‘location’ estimation (very) sensitive to extreme data

- data contamination (e.g., ‘decimal point’ encoding error’)?
- ‘valid’ outliers?

Consequences are potential bias and high sampling uncertainty (even with large samples).

$\Longrightarrow$ Many measures of interest have ‘unbounded influence function’

The problem of data contamination and extreme values

The problem

Analysis beyond ‘central tendency’/‘location’ estimation (very) sensitive to extreme data

- data contamination (e.g., ‘decimal point’ encoding error’)?
- ‘valid’ outliers?

Consequences are potential bias and high sampling uncertainty (even with large samples).

$\Longrightarrow$ Many measures of interest have ‘unbounded influence function’

The problem of data contamination and extreme values

The problem

Analysis beyond 'central tendency'/'location' estimation (very) sensitive to extreme data

- data contamination (e.g., 'decimal point' encoding error')?
- 'valid' outliers?

Consequences are potential bias and high sampling uncertainty (even with large samples).

$\Longrightarrow$ Many measures of interest have 'unbounded influence function'

The problem of data contamination and extreme values

The problem

Analysis beyond 'central tendency'/'location' estimation (very) sensitive to extreme data

- data contamination (e.g., 'decimal point' encoding error')?
- 'valid' outliers?

Consequences are potential bias and high sampling uncertainty (even with large samples).

$\Longrightarrow$ Many measures of interest have 'unbounded influence function'

Influence function examples – Inequality indices

from Cowell & Flachaire (2007)

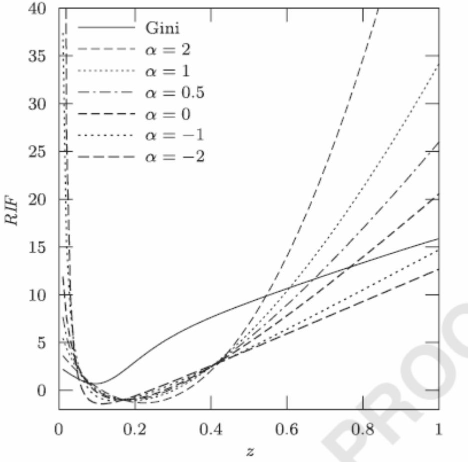
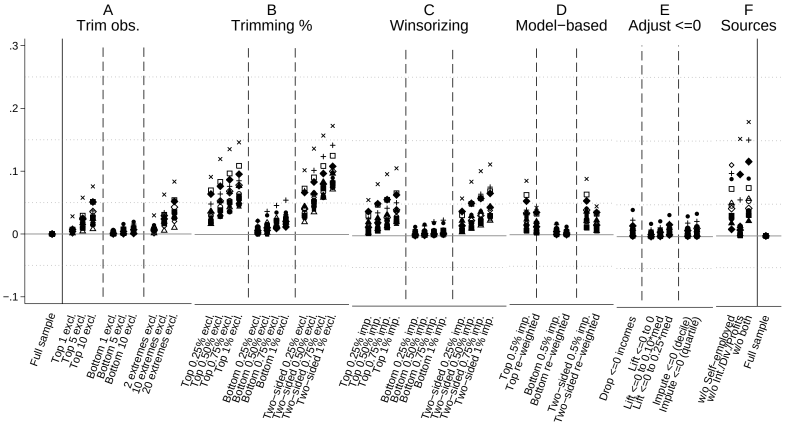


Fig. 1. *IF*s of generalised entropy I_E^α.

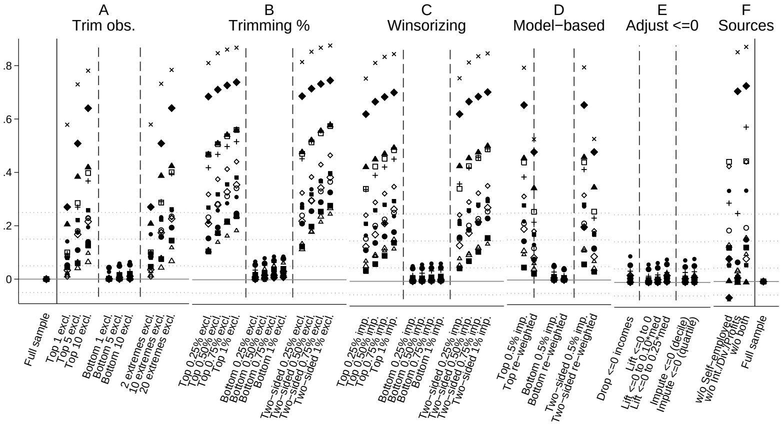
Impact of extreme incomes adjustments – Gini

from Van Kerm (2007)



Extreme incomes adjustments – GE(2)

from Van Kerm (2007)



[outline]

Remedial actions

1. Identify and adjust extreme data: removal, re-coding
   - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
   - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
   - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
   - model only the tails of the distribution parametrically (e.g. Pareto)
   - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

Remedial actions

1. Identify and adjust extreme data: removal, re-coding
   - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
   - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
   - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
   - model only the tails of the distribution parametrically (e.g. Pareto)
   - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

Remedial actions

1. Identify and adjust extreme data: removal, re-coding
   - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
   - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
   - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
   - model only the tails of the distribution parametrically (e.g. Pareto)
   - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

Remedial actions

1. Identify and adjust extreme data: removal, re-coding
   - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
   - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
   - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
   - model only the tails of the distribution parametrically (e.g. Pareto)
   - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

Remedial actions

1. Identify and adjust extreme data: removal, re-coding
   - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
   - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
   - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
   - model only the tails of the distribution parametrically (e.g. Pareto)
   - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

Remedial actions

1. Identify and adjust extreme data: removal, re-coding
   - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
   - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
   - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
   - model only the tails of the distribution parametrically (e.g. Pareto)
   - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

Remedial actions

1. Identify and adjust extreme data: removal, re-coding
   - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
   - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
   - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
   - model only the tails of the distribution parametrically (e.g. Pareto)
   - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

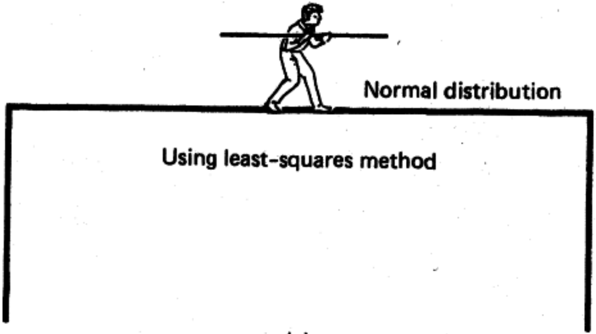
Remedial actions

1. Identify and adjust extreme data: removal, re-coding
    - Relatively easy, but not efficient and dependence to ad-hoc trimming fractions
    - Impact can be substantial ... and difficult to justify
2. Rely on functional form assumptions:
    - model the full distribution parametrically (e.g. log-Normal, Gamma), so distribution fully characterized by just a few parameters
    - model only the tails of the distribution parametrically (e.g. Pareto)
    - But... classical ML estimators of distribution parameters are themselves non-robust to extreme values!

$\Longrightarrow$ Solution discussed in this talk: **Use "robust" estimators of model parameters** (instead of classical ML)

Robust estimation methods

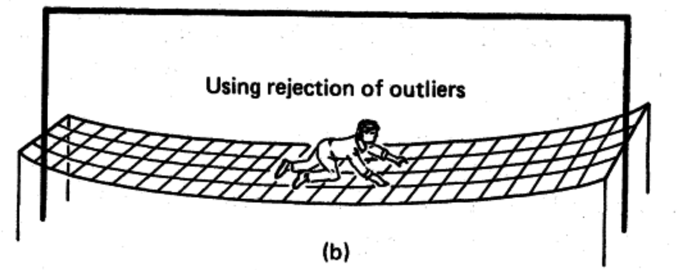
(Hampel , 1986)



(a)

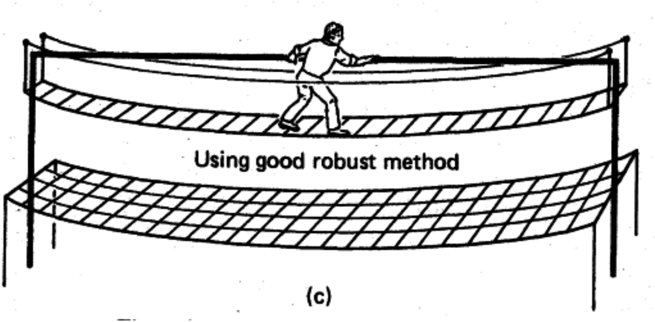
Robust estimation methods

(Hampel , 1986)



Robust estimation methods

(Hampel , 1986)



The estimation problem

Task

We want to fit a given parametric distribution f_θ to the available data: θ is a vector of parameters to be estimated.

ML estimation

Find θ^{ML} solution to $\sum_{i=1}^{N} s(x_i, \theta^{ML}) = 0$, where $s(x_i, \theta^{ML})$ is the score function: $s(x_i, \theta) = \partial \log(f_\theta(x_i))/\partial\theta$

Problem

The score function has unbounded influence function for almost all classic models of size distributions. Parameter estimates can therefore be driven to arbitrary values by data contamination...

The estimation problem

Task

We want to fit a given parametric distribution f_θ to the available data: θ is a vector of parameters to be estimated.

ML estimation

Find θ^{ML} solution to $\sum_{i=1}^{N} s(x_i, \theta^{ML}) = 0$, where $s(x_i, \theta^{ML})$ is the score function: $s(x_i, \theta) = \partial \log(f_\theta(x_i))/\partial\theta$

Problem

The score function has unbounded influence function for almost all classic models of size distributions. Parameter estimates can therefore be driven to arbitrary values by data contamination...

The estimation problem

Task

We want to fit a given parametric distribution f_θ to the available data: θ is a vector of parameters to be estimated.

ML estimation

Find θ^{ML} solution to $\sum_{i=1}^{N} s(x_i, \theta^{ML}) = 0$, where $s(x_i, \theta^{ML})$ is the score function: $s(x_i, \theta) = \partial \log(f_\theta(x_i))/\partial\theta$

Problem

The score function has unbounded influence function for almost all classic models of size distributions. Parameter estimates can therefore be driven to arbitrary values by data contamination...

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

OBRE

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

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

where

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

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

OBRE

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

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

where

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

Optimal B-Robust Estimators (OBRE)

A robust alternative to classical ML

OBRE

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

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

where

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

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

A robust alternative to classical ML

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

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

A robust alternative to classical ML

- $W_c(x; \theta^{OB})$ imposes a bound on influence function by downweighting extreme values (values deviating from model)

- c is a 'robustness' parameter to be determined ex ante (tune efficiency-robustness trade-off)
    - If $c \to \infty$ then $\theta^{OB} = \theta^{ML}$

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

A robust alternative to classical ML

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

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

A robust alternative to classical ML

- $a(\theta^{OB})$ and $A(\theta^{OB})$ are implicitly defined as

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

$\Longrightarrow$ The resulting estimator is the optimal (minimum variance) M-estimator with bounded influence function

- For a thorough discussion, 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 implicitly defined as

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

$\Longrightarrow$ The resulting estimator is the optimal (minimum variance) M-estimator with bounded influence function

- For a thorough discussion, 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 implicitly defined as

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

$\Longrightarrow$ The resulting estimator is the optimal (minimum variance) M-estimator with bounded influence function

- For a thorough discussion, 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 implemented Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm:
  - given some θ, solve equations for $a(\theta)$ and $A(\theta)$
  - with new $a(\theta)$ and $A(\theta)$, determine new $W_c(x_i;\theta)$ and update θ (Newton-Raphson step) until convergence
- Solving equations for $a(\theta)$ and $A(\theta)$ also based on an iterative procedure

$\Longrightarrow$ Rather difficult problem, and very computer-intensive (esp. for numerical integration). So needs

1. speed
2. matrix operations

$\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I implemented Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm:
  - given some θ, solve equations for $a(\theta)$ and $A(\theta)$
  - with new $a(\theta)$ and $A(\theta)$, determine new $W_c(x_i;\theta)$ and update θ (Newton-Raphson step) until convergence
- Solving equations for $a(\theta)$ and $A(\theta)$ also based on an iterative procedure

$\Longrightarrow$ Rather difficult problem, and very computer-intensive (esp. for numerical integration). So needs

1. speed
2. matrix operations

$\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I implemented Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm:
  - given some θ, solve equations for $a(\theta)$ and $A(\theta)$
  - with new $a(\theta)$ and $A(\theta)$, determine new $W_c(x_i;\theta)$ and update θ (Newton-Raphson step) until convergence
- Solving equations for $a(\theta)$ and $A(\theta)$ also based on an iterative procedure

$\Longrightarrow$ Rather difficult problem, and very computer-intensive (esp. for numerical integration). So needs

1. speed
2. matrix operations

$\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I implemented Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm:
  - given some θ, solve equations for $a(\theta)$ and $A(\theta)$
  - with new $a(\theta)$ and $A(\theta)$, determine new $W_c(x_i;\theta)$ and update θ (Newton-Raphson step) until convergence
- Solving equations for $a(\theta)$ and $A(\theta)$ also based on an iterative procedure

$\Longrightarrow$ Rather difficult problem, and very computer-intensive (esp. for numerical integration). So needs

1. speed
2. matrix operations

$\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I implemented Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm:
  - given some θ, solve equations for $a(\theta)$ and $A(\theta)$
  - with new $a(\theta)$ and $A(\theta)$, determine new $W_c(x_i;\theta)$ and update θ (Newton-Raphson step) until convergence
- Solving equations for $a(\theta)$ and $A(\theta)$ also based on an iterative procedure

$\Longrightarrow$ Rather difficult problem, and very computer-intensive (esp. for numerical integration). So needs

1. speed
2. matrix operations

$\Longrightarrow$ Mata!

Implementation

- Given number of implicit definitions of parameters and constraints, estimation is not easy
- But relatively detailed algorithms are available (fortunately!). I implemented Ronchetti & Victoria-Feser (*Canadian Journal of Statistics*, 1994).
- Iterative algorithm:
  - given some θ, solve equations for $a(\theta)$ and $A(\theta)$
  - with new $a(\theta)$ and $A(\theta)$, determine new $W_c(x_i;\theta)$ and update θ (Newton-Raphson step) until convergence
- Solving equations for $a(\theta)$ and $A(\theta)$ also based on an iterative procedure

$\Longrightarrow$ Rather difficult problem, and very computer-intensive (esp. for numerical integration). So needs

1. speed
2. matrix operations

$\Longrightarrow$ Mata!

Implementation (ctd.)

- Implementation is “relatively easy” with Mata (but familiarity with matrix algebra can help!)
- Uses a suite of existing commands by Stephen Jenkins to fit functional forms to unit record data by ML
  - just replace ML engine by home-brewed OBRE engine
  - i.e. call a Mata function, rather than `ml model`!

    ```
    void gamma_obre(string scalar varname, string
    scalar sweight, string scalar touse, string
    scalar thenewvar, real scalar froma, real scalar
    fromb , real scalar c)
    ```
  - the Mata function return a vector of parameter estimates along with a covariance matrix estimate
- To date I implemented Pareto Type I (1 param), log-Normal and Gamma (2 params) and Singh-Maddala (3 params)
- Compatible with Nick Cox’s diagnostic commands `p*` and `q*` for pp-plot and qq-plot

Implementation (ctd.)

- Implementation is “relatively easy” with Mata (but familiarity with matrix algebra can help!)
- Uses a suite of existing commands by Stephen Jenkins to fit functional forms to unit record data by ML
  - just replace ML engine by home-brewed OBRE engine
  - i.e. call a Mata function, rather than `ml model`!

    ```
    void gamma_obre(string scalar varname, string
    scalar sweight, string scalar touse, string
    scalar thenewvar, real scalar froma, real scalar
    fromb , real scalar c)
    ```
  - the Mata function return a vector of parameter estimates along with a covariance matrix estimate
- To date I implemented Pareto Type I (1 param), log-Normal and Gamma (2 params) and Singh-Maddala (3 params)
- Compatible with Nick Cox’s diagnostic commands `p*` and `q*` for pp-plot and qq-plot

Implementation (ctd.)

- Implementation is "relatively easy" with Mata (but familiarity with matrix algebra can help!)
- Uses a suite of existing commands by Stephen Jenkins to fit functional forms to unit record data by ML
  - just replace ML engine by home-brewed OBRE engine
  - i.e. call a Mata function, rather than `ml model`!

    ```
    void gamma_obre(string scalar varname, string scalar sweight, string scalar touse, string scalar thenewvar, real scalar froma, real scalar fromb , real scalar c)
    ```
  - the Mata function return a vector of parameter estimates along with a covariance matrix estimate
- To date I implemented Pareto Type I (1 param), log-Normal and Gamma (2 params) and Singh-Maddala (3 params)
- Compatible with Nick Cox's diagnostic commands `p*` and `q*` for pp-plot and qq-plot

Implementation (ctd.)

- Implementation is "relatively easy" with Mata (but familiarity with matrix algebra can help!)
- Uses a suite of existing commands by Stephen Jenkins to fit functional forms to unit record data by ML
  - just replace ML engine by home-brewed OBRE engine
  - i.e. call a Mata function, rather than `ml model`!

    ```
    void gamma_obre(string scalar varname, string
    scalar sweight, string scalar touse, string
    scalar thenewvar, real scalar froma, real scalar
    fromb , real scalar c)
    ```
  - the Mata function return a vector of parameter estimates along with a covariance matrix estimate
- To date I implemented Pareto Type I (1 param), log-Normal and Gamma (2 params) and Singh-Maddala (3 params)
- Compatible with Nick Cox's diagnostic commands `p*` and `q*` for pp-plot and qq-plot

Implementation (ctd.)

- Implementation is "relatively easy" with Mata (but familiarity with matrix algebra can help!)
- Uses a suite of existing commands by Stephen Jenkins to fit functional forms to unit record data by ML
  - just replace ML engine by home-brewed OBRE engine
  - i.e. call a Mata function, rather than `ml model`!

    ```
    void gamma_obre(string scalar varname, string
    scalar sweight, string scalar touse, string
    scalar thenewvar, real scalar froma, real scalar
    fromb , real scalar c)
    ```
  - the Mata function return a vector of parameter estimates along with a covariance matrix estimate
- To date I implemented Pareto Type I (1 param), log-Normal and Gamma (2 params) and Singh-Maddala (3 params)
- Compatible with Nick Cox's diagnostic commands `p*` and `q*` for pp-plot and qq-plot

Implementation (ctd.)

- Implementation is "relatively easy" with Mata (but familiarity with matrix algebra can help!)
- Uses a suite of existing commands by Stephen Jenkins to fit functional forms to unit record data by ML
  - just replace ML engine by home-brewed OBRE engine
  - i.e. call a Mata function, rather than `ml model`!

    ```
    void gamma_obre(string scalar varname, string
    scalar sweight, string scalar touse, string
    scalar thenewvar, real scalar froma, real scalar
    fromb , real scalar c)
    ```
  - the Mata function return a vector of parameter estimates along with a covariance matrix estimate
- To date I implemented Pareto Type I (1 param), log-Normal and Gamma (2 params) and Singh-Maddala (3 params)
- Compatible with Nick Cox's diagnostic commands `p*` and `q*` for pp-plot and qq-plot

Implementation (ctd.)

- Implementation is "relatively easy" with Mata (but familiarity with matrix algebra can help!)
- Uses a suite of existing commands by Stephen Jenkins to fit functional forms to unit record data by ML
  - just replace ML engine by home-brewed OBRE engine
  - i.e. call a Mata function, rather than `ml model`!

    ```
    void gamma_obre(string scalar varname, string
    scalar sweight, string scalar touse, string
    scalar thenewvar, real scalar froma, real scalar
    fromb , real scalar c)
    ```
  - the Mata function return a vector of parameter estimates along with a covariance matrix estimate
- To date I implemented Pareto Type I (1 param), log-Normal and Gamma (2 params) and Singh-Maddala (3 params)
- Compatible with Nick Cox's diagnostic commands `p*` and `q*` for pp-plot and qq-plot

Practical programming issues

- Precision of numerical integration functions is important...
- ... and drives estimation speed
- 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 is important...
- ... and drives estimation speed
- 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 is important...
- ... and drives estimation speed
- 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 is important...
- ... and drives estimation speed
- 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, $\theta > 0$)

Output

```
Starting values (ML estimates): [ a = 4.430 ; b = 589.051 ]
Estimation with OBRE robustness constant set to c = 5

Iteration  1:  (....................) a = 5.116, b = 492.598
Iteration  2:  (....................) a = 5.366, b = 467.565
Iteration  3:  (....................) a = 5.466, b = 457.468
Iteration  4:  (..................) a = 5.516, b = 452.570
Iteration  5:  (................) a = 5.542, b = 450.096
Iteration  6:  (..............) a = 5.556, b = 448.812
Iteration  7:  (............) a = 5.563, b = 448.137
Iteration  8:  (..........)   a = 5.567, b = 447.780
Iteration  9:  (........)     a = 5.569, b = 447.591
Iteration 10:  (......)       a = 5.570, b = 447.490
Iteration 11:  (....)         a = 5.571, b = 447.435
Iteration 12:  (...)          a = 5.571, b = 447.407
Iteration 13:  (..)           a = 5.571, b = 447.391
Iteration 14:  (.)            a = 5.571, b = 447.383
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
a						
_cons	5.571198	.0580081	96.04	0.000	5.457504	5.684891
b						
_cons	447.3829	4.091696	109.34	0.000	439.3633	455.4025

Half CV^2	.089747
Gini coeff.	.23373
Theil	.087071

[outline]

Set-up

Monte Carlo simulation

1. Draw samples from known distributions
2. Add various kind of contamination – decimal point error – to a fraction of sample data
3. Estimate parameters from datasets using both ML and OBRE

- Pareto with sample size of 200
- log-Normal and Singh-Maddala with samples of size 1000

Set-up

Monte Carlo simulation

1. Draw samples from known distributions
2. Add various kind of contamination – decimal point error – to a fraction of sample data
3. Estimate parameters from datasets using both ML and OBRE

- Pareto with sample size of 200
- log-Normal and Singh-Maddala with samples of size 1000

Set-up (ctd.)

Types of contamination

1. 1% of obs. multiplied by 10
2. 1% of obs. divided by 10
3. 1% of obs. mulitplied by 10 and 1% of obs. divided by 10
4. 3% of obs. multiplied by 10
5. 3% of obs. divided by 10

Results

Pareto distribution

True parameter value: $\alpha = 3$

Model	root MSE		
	ML	c=5	c=2
No cont.	0.215	0.214	0.230
1% *10	0.261	0.252	0.231
3% *10	0.527	0.521	0.286

Results

log-Normal distribution

Model	Param.		root MSE		
			ML	c=5	c=3
No cont.	μ	8	0.017	0.017	0.017
	σ	.525	0.012	0.013	0.031
	Gini	0.290	0.006	0.007	0.017
	Theil	0.138	0.006	0.007	0.016
	$.5CV^2$	0.159	0.008	0.009	0.020
1% *10	μ	8	0.029	0.020	0.018
	σ	.525	0.050	0.020	0.021
	Gini	0.290	0.026	0.011	0.011
	Theil	0.138	0.027	0.011	0.011
	$.5CV^2$	0.159	0.037	0.014	0.014

Results

log-Normal distribution (ctd.)

Model	Param.	True	root MSE		
			ML	c=5	c=3
3% *10	μ	8	0.072	0.043	0.025
	σ	.525	0.131	0.070	0.016
	Gini	0.290	0.068	0.037	0.008
	Theil	0.138	0.078	0.040	0.009
	$.5CV^2$	0.159	0.111	0.054	0.011
3% /10	μ	8	0.070	0.047	0.025
	σ	.525	0.132	0.082	0.017
	Gini	0.290	0.068	0.043	0.009
	Theil	0.138	0.078	0.046	0.009
	$.5CV^2$	0.159	0.111	0.064	0.012

Results

Singh-Maddala distribution

Model	Param.	True	root MSE		
			ML	c=7	c=5
No cont.	α	2.8	0.128	0.145	0.301
	β	3500	297	283	590
	p	1.7	0.283	0.252	0.522
	Gini	0.289	0.008	0.009	0.016
	Theil	0.132	0.016	0.014	0.030
	$.5CV^2$	0.162	0.016	0.020	0.059
1% *10	α	2.8	0.297	0.243	0.370
	β	3500	720	572	751
	p	1.7	0.652	0.519	0.665
	Gini	0.289	0.032	0.021	0.027
	Theil	0.132	0.026	0.025	0.024
	$.5CV^2$	0.162	0.118	0.071	0.109

Results

Singh-Maddala distribution (ctd.)

Model	Param.	True	root MSE		
			ML	c=5	c=3
3% $\times$10	α	2.8	0.511	0.472	0.494
	β	3500	1145	1069	1004
	p	1.7	0.991	0.935	0.880
	Gini	0.289	0.088	0.073	0.055
	Theil	0.132	0.245	0.160	0.107
	$.5CV^2$	0.162	1.154	0.547	0.320
3% /10	α	2.8	0.578	0.521	0.253
	β	3500	1814	1306	788
	p	1.7	1.859	1.309	0.869
	Gini	0.289	0.022	0.021	0.021
	Theil	0.132	172.324	0.586	3.030
	$.5CV^2$	0.162	0.014	0.015	0.036

Main observations

- OBRE very useful with Pareto and, especially, log-Normal models
- OBRE useful too with Singh-Maddala, yet
  - choice of c matter – too much robustness not good with small contamination
  - too much contamination remains very harmful (look at impact on estimates of 'sensitive' inequality measures (Theil, $.5CV^2$)!) – even with OBRE
- Convergence problems with Gamma models – otherwise results similar to SM

Main observations

- OBRE very useful with Pareto and, especially, log-Normal models
- OBRE useful too with Singh-Maddala, yet
  - choice of c matter – too much robustness not good with small contamination
  - too much contamination remains very harmful (look at impact on estimates of 'sensitive' inequality measures (Theil, $.5CV^2$)!) – even with OBRE
- Convergence problems with Gamma models – otherwise results similar to SM

Main observations

- OBRE very useful with Pareto and, especially, log-Normal models
- OBRE useful too with Singh-Maddala, yet
  - choice of c matter – too much robustness not good with small contamination
  - too much contamination remains very harmful (look at impact on estimates of 'sensitive' inequality measures (Theil, $.5CV^2$)!) – even with OBRE
- Convergence problems with Gamma models – otherwise results similar to SM

Main observations

- OBRE very useful with Pareto and, especially, log-Normal models
- OBRE useful too with Singh-Maddala, yet
    - choice of c matter – too much robustness not good with small contamination
    - too much contamination remains very harmful (look at impact on estimates of ‘sensitive’ inequality measures (Theil, .5CV^2)!) – even with OBRE
- Convergence problems with Gamma models – otherwise results similar to SM

Main observations

- OBRE very useful with Pareto and, especially, log-Normal models
- OBRE useful too with Singh-Maddala, yet
  - choice of c matter – too much robustness not good with small contamination
  - too much contamination remains very harmful (look at impact on estimates of 'sensitive' inequality measures (Theil, $.5CV^2$)!) – even with OBRE
- Convergence problems with Gamma models – otherwise results similar to SM

[outline]

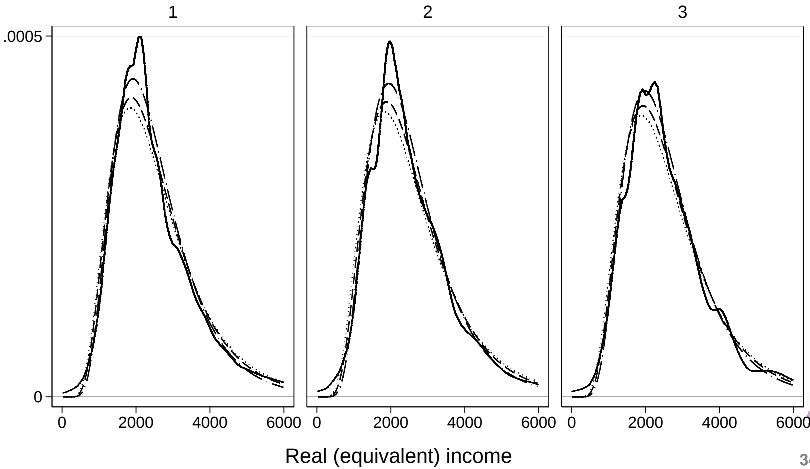
Data

PSELL-III

- Panel Survey “Liewen zu Letzebuerg”, waves 1(2003)-3(2005)
- Representative of residents in Luxembourg
- Real annual household income (in single adult equivalent)

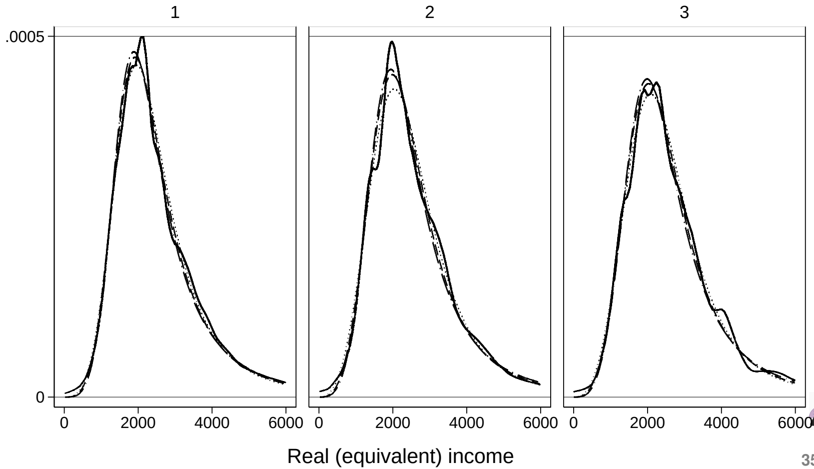
PDF estimates for log-Normal fit

OBRE improves fit, but not very good model



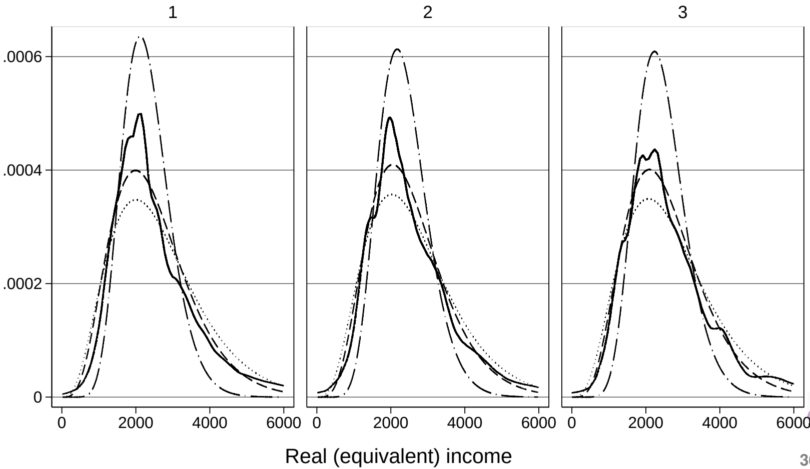
PDF estimates for Singh-Maddala fit

OBRE useful and much better fit

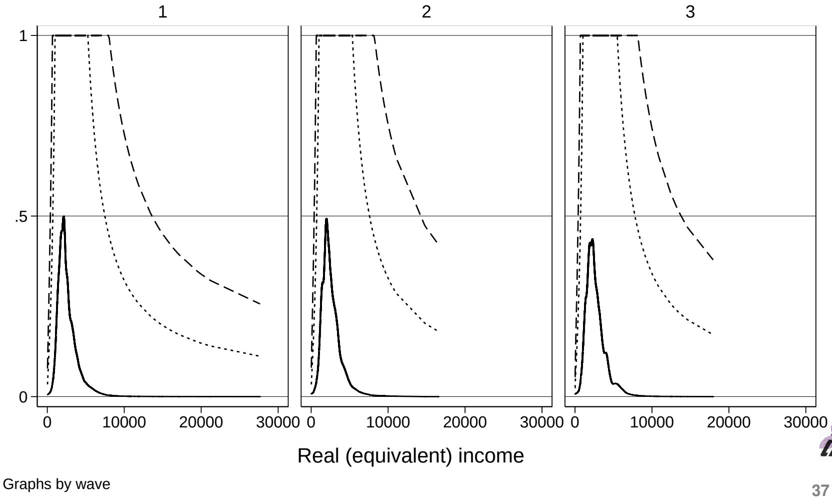


PDF estimates for Gamma fit

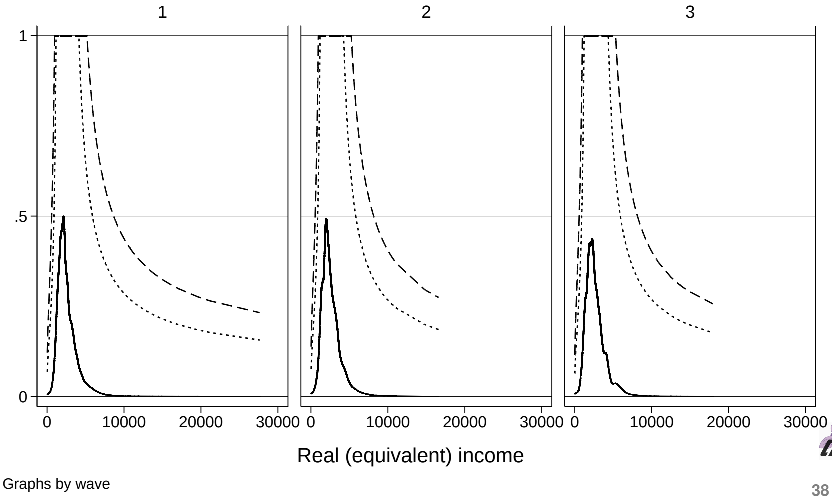
(does it call for any comment?)



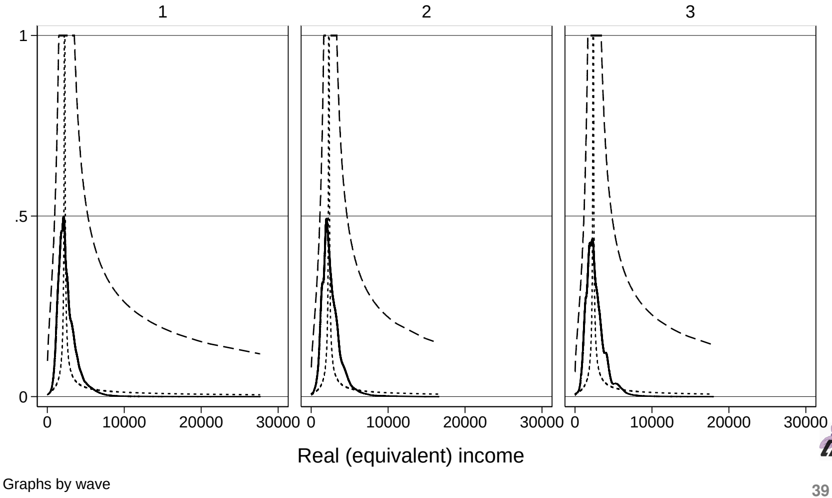
OBRE weights for log-Normal fit



OBRE weights for Singh-Maddala fit



OBRE weights for Gamma fit



[outline]

The principle

- More flexible approach is to focus on distribution tails
  - bulk of the data are taken at face value – use empirical CDF
  - parametric approach only for the tails – largest (and smallest?) observations are used to estimate a parametric model
  - empirical CDF combined with parametric CDFs for estimation of, say, inequality measures, stochastic dominance, etc.
- Under assumption that the CDF "decays as a power function" – i.e., has a heavy tail –, fitting a Pareto distribution to tail data is a valid choice: for $x \geq z$,

$$F(x) = 1 - \left(\frac{x}{z}\right)^{-\alpha}$$

The principle

- More flexible approach is to focus on distribution tails
  - bulk of the data are taken at face value – use empirical CDF
  - parametric approach only for the tails – largest (and smallest?) observations are used to estimate a parametric model
  - empirical CDF combined with parametric CDFs for estimation of, say, inequality measures, stochastic dominance, etc.
- Under assumption that the CDF "decays as a power function" – i.e., has a heavy tail –, fitting a Pareto distribution to tail data is a valid choice: for $x \geq z$,

$$F(x) = 1 - \left(\frac{x}{z}\right)^{-\alpha}$$

The principle

- More flexible approach is to focus on distribution tails
  - bulk of the data are taken at face value – use empirical CDF
  - parametric approach only for the tails – largest (and smallest?) observations are used to estimate a parametric model
  - empirical CDF combined with parametric CDFs for estimation of, say, inequality measures, stochastic dominance, etc.
- Under assumption that the CDF "decays as a power function" – i.e., has a heavy tail –, fitting a Pareto distribution to tail data is a valid choice: for $x \geq z$,

$$F(x) = 1 - \left(\frac{x}{z}\right)^{-\alpha}$$

Pareto tail estimation

- OBRE estimator useful to avoid influence of contamination on Pareto parameter estimate α
- Main issue is the choice of z – value beyond which data are modelled parametrically

  $\Longrightarrow$ Pareto quantile plot and Hill's plot
  - Under Pareto model, linear relationship between $-\log(1 - F(x))$ and $\log(x)$ – so help detecting reasonable value of z
  - (yet difficulty associated with contamination at the very top)

Pareto tail estimation

- OBRE estimator useful to avoid influence of contamination on Pareto parameter estimate α
- Main issue is the choice of z – value beyond which data are modelled parametrically

$\Longrightarrow$ Pareto quantile plot and Hill's plot

- Under Pareto model, linear relationship between $-\log(1 - F(x))$ and $\log(x)$ – so help detecting reasonable value of z
- (yet difficulty associated with contamination at the very top)

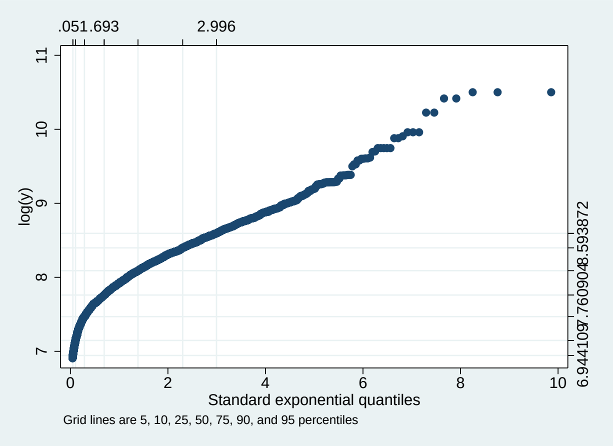
Pareto tail estimation

- OBRE estimator useful to avoid influence of contamination on Pareto parameter estimate α
- Main issue is the choice of z – value beyond which data are modelled parametrically

$\Longrightarrow$ Pareto quantile plot and Hill's plot

  - Under Pareto model, linear relationship between $-\log(1 - F(x))$ and $\log(x)$ – so help detecting reasonable value of z
  - (yet difficulty associated with contamination at the very top)

Pareto quantile plot



(Stata command `pareto_logqplot` available in package for Pareto tail modelling – coming soon on SSC!)

Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is (still) a prototype (but looks ok). Minor developments still needed for
  - fixing precision and tolerance thresholds
  - additional distributions (GB2?) – transplanting code to other distributions is easy, yet more convergence problems to be expected with higher number of parameters

Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is (still) a prototype (but looks ok). Minor developments still needed for
    - fixing precision and tolerance thresholds
    - additional distributions (GB2?) – transplanting code to other distributions is easy, yet more convergence problems to be expected with higher number of parameters

Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is (still) a prototype (but looks ok). Minor developments still needed for
  - fixing precision and tolerance thresholds
  - additional distributions (GB2?) – transplanting code to other distributions is easy, yet more convergence problems to be expected with higher number of parameters

Concluding remarks

- Mata makes estimators such as OBRE feasible within Stata
- In theory, OBRE estimators have great relevance in distribution analysis... implementation in Stata may help putting this claim to broader practical assessment
- At present, it is (still) a prototype (but looks ok). Minor developments still needed for
  - fixing precision and tolerance thresholds
  - additional distributions (GB2?) – transplanting code to other distributions is easy, yet more convergence problems to be expected with higher number of parameters

Cowell, F. A. & Flachaire, E. (2007), 'Income distribution and inequality measurement: The problem of extreme values', *Journal of Econometrics*, doi:10.1016/j.jeconom.2007.01.001 (forthcoming).

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J. & Stahel, W. A. (1986), *Robust statistics: The approach based on influence functions*, John Wiley, New York.

Van Kerm, P. (2007), 'Extreme incomes and the estimation of poverty and inequality indicators from EU-SILC', IRISS Working Paper 2007-01, CEPS/INSTEAD, Differdange, Luxembourg.