The ineq Package

January 24, 2005

Version 0.2-5

Date 2005-01-21

Title Measuring inequality, concentration and poverty

Author Achim Zeileis

Maintainer Achim Zeileis <Achim.Zeileis@R-project.org>

Description Inequality, concentration and poverty measures Lorenz curves (empirical and theoretical)

License GPL

R topics documented:

`Ilocos` *Income Metadata from Ilocos, Philippines*

Description

Income metadata from surveys conducted by the Philippines' National Statistics Office.

Usage

```
data(Ilocos)
```

Format

A data frame with 632 observations of 8 variables.

income total income of household,

sex sex of household head (`"male"` or `"female"`),

family.size family size (sometimes averaged over two semesters),

urbanity factor with levels `"rural"` and `"urban"`,

province factor indicating the particular province,

AP.income total household income during the APIS,

AP.family.size family size during APIS,

AP.weight APIS survey weight for each household.

Details

The data contains household income and metadata in one of the sixteen regions of the Philippines called Ilocos. The data comes from two of the NSO's surveys: the 1997 Family and Income and Expenditure Survey and the 1998 Annual Poverty Indicators Survey (APIS).

Since the APIS only has a six month reference period, the original data were rescaled using an adjustment factor from the quarterly GDP figures that can be obtained from the major sectors.

Source

National Statistics Office, Philippines: `http://www.census.gov.ph/`, where also the whole data set may be obtained.

Lc *Lorenz Curve*

Description

Computes the (empirical) ordinary and generalized Lorenz curve of a vector x

Usage

```
Lc(x, n = rep(1,length(x)), plot = FALSE)
```

Arguments

`x`	a vector containing non-negative elements.
`n`	a vector of frequencies, must be same length as `x`.
`plot`	logical. If TRUE the empirical Lorenz curve will be plotted.

Details

`Lc(x)` computes the empirical ordinary Lorenz curve of `x` as well as the generalized Lorenz curve (= ordinary Lorenz curve * mean(x)). The result can be interpreted like this: `p`*100 percent have `L(p)`*100 percent of `x`.

If `n` is changed to anything but the default `x` is interpreted as a vector of class means and `n` as a vector of class frequencies: in this case `Lc` will compute the minimal Lorenz curve (= no inequality within each group). A maximal curve can be computed with `Lc.mehran`.

Value

A list of class `"Lc"` with the following components:

`p`	vector of percentages
`L`	vector with values of the ordinary Lorenz curve
`L.general`	vector with values of the generalized Lorenz curve

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

B C Arnold: Majorization and the Lorenz Order: A Brief Introduction, 1987, Springer,

F A Cowell: Measurement of Inequality, 2000, in A B Atkinson / F Bourguignon (Eds): Handbook of Income Distribution, Amsterdam,

F A Cowell: Measuring Inequality, 1995 Prentice Hall/Harvester Wheatshef.

See Also

`plot.Lc`, `Lc.mehran`, `plot.theorLc`

Examples

```
## Load and attach income (and metadata) set from Ilocos, Philippines
data(Ilocos)
attach(Ilocos)
## extract and rescale income for the provinces "Pangasinan" und "La Union"
income.p <- income[province=="Pangasinan"]/10000
income.u <- income[province=="La Union"]/10000

## compute the Lorenz curves
Lc.p <- Lc(income.p)
Lc.u <- Lc(income.u)
## it can be seen the the inequality in La Union is higher than in
## Pangasinan because the respective Lorenz curve takes smaller values.
plot(Lc.p)
lines(Lc.u, col=2)
## the picture becomes even clearer with generalized Lorenz curves
plot(Lc.p, general=TRUE)
lines(Lc.u, general=TRUE, col=2)
## inequality measures emphasize these results, e.g. Atkinson's measure
ineq(income.p, type="Atkinson")
ineq(income.u, type="Atkinson")
## or Theil's entropy measure
ineq(income.p, type="Theil", parameter=0)
ineq(income.u, type="Theil", parameter=0)


# income distribution of the USA in 1968 (in 10 classes)
# x vector of class means, n vector of class frequencies
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
n <- c(482, 825, 722, 690, 661, 760, 745, 2140, 1911, 1024)
```

```
# compute minimal Lorenz curve (= no inequality in each group)
Lc.min <- Lc(x, n=n)
# compute maximal Lorenz curve (limits of Mehran)
Lc.max <- Lc.mehran(x,n)
# plot both Lorenz curves in one plot
plot(Lc.min)
lines(Lc.max, col=4)

# add the theoretic Lorenz curve of a Lognormal-distribution with variance 0.78
lines(Lc.lognorm, parameter=0.78)
# add the theoretic Lorenz curve of a Dagum-distribution
lines(Lc.dagum, parameter=c(3.4,2.6))
```

`Lc.mehran` *Mehran Bounds For Lorenz Curves*

Description

Computes the Mehran bounds for a Lorenz curve of grouped data

Usage

```
Lc.mehran(x,n)
```

Arguments

`x`	vector of class means.
`n`	vector of class frequencies.

Value

An object of class `"Lc"`, but containing only `p` and `L`.

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

F Mehran: Bounds on the Gini Index Based on Observed Points of the Lorenz Curve, 1975, JASA 70, 64-66.

See Also

`Lc`, `plot.Lc`, `plot.theorLc`

Examples

```
# income distribution of the USA in 1968 (in 10 classes)
# x vector of class means, n vector of class frequencies
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
n <- c(482, 825, 722, 690, 661, 760, 745, 2140, 1911, 1024)

# compute minimal Lorenz curve (= no inequality in each group)
Lc.min <- Lc(x, n=n)
# compute maximal Lorenz curve (limits of Mehran)
Lc.max <- Lc.mehran(x,n)
# plot both Lorenz curves in one plot
plot(Lc.min)
lines(Lc.max, col=4)

# add the theoretic Lorenz curve of a Lognormal-distribution with variance 0.78
lines(Lc.lognorm, parameter=0.78)
# add the theoretic Lorenz curve of a Dagum-distribution
lines(Lc.dagum, parameter=c(3.4,2.6))
```

Pen *Pen's Parade*

Description

plots Pen's Parade of a vector `x`

Usage

```
Pen(x, n = rep(1, length(x)), group = NULL,
  scaled = TRUE, abline = TRUE, segments = NULL,
  main = "Pen's Parade", ylab = NULL, xlab = NULL,
  col = NULL, lwd = NULL, las = 1, fill = NULL, ...)
```

Arguments

`x`	a vector containing non-negative elements.
`n`	a vector of frequencies or weights, must be same length as `x`.
`group`	a factor coding different groups, must be same length as `x`. See also details.
`scaled`	logical. Should Pen's parade be divided by `mean(x)`?
`abline`	logical. Should a horizontal line for the mean be drawn?
`segments`	logical. Should histogram-like segments be drawn?
`col`	a (vector of) color(s) for drawing the curve.
`fill`	a (vector of) color(s) for filling the area under the curve.
`xlab,ylab`	axis labels. Suitable defaults depending on `scaled` and `n` are chosen.
`main, lwd, las, ...`	further high-level `plot` parameters.

Details

Pen's Parade is basically the inverse distribution function (standardized by `mean(x)`).

`Pen` allows for fine control of the layout—the graphical parameters `col` and `fill` can be vectorized if histogram-like segments are drawn (`segments = TRUE`)—but implements several heuristics in choosing its default plotting parameters. If a grouping factor `group` is given, the default is to draw segments with a grey-shaded filling. If no fill color is used, the default is to draw a thick blue curve. But as all of these are just defaults, they can of course easily be changed. See also the examples.

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

F A Cowell: Measurement of Inequality, 2000, in A B Atkinson / F Bourguignon (Eds): Handbook of Income Distribution, Amsterdam,

F A Cowell: Measuring Inequality, 1995 Prentice Hall/Harvester Wheatshef,

J Pen: Income Distribution, 1971, Harmondsworth: Allen Lane.

See Also

`Lc`, `plot.Lc`

Examples

```
# load and attach Philippine income data
data(Ilocos)
attach(Ilocos)
# plot Pen's Parade of income
Pen(income)
Pen(income, fill = hsv(0.1, 0.3, 1))

# income distribution of the USA in 1968 (in 10 classes)
# x vector of class means, n vector of class frequencies
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
n <- c(482, 825, 722, 690, 661, 760, 745, 2140, 1911, 1024)
Pen(x, n = n)
# create artificial grouping variable
myfac <- factor(c(1, 1, 1, 2, 2, 2, 3, 3, 3, 3))
Pen(x, n = n, group = myfac)
```

---

conc *Concentration Measures*

Description

computes the concentration within a vector according to the specified concentration measure

Usage

```
conc(x, parameter = NULL, type = c("Herfindahl", "Rosenbluth"))

Herfindahl(x, parameter = 1)
Rosenbluth(x)
```

Arguments

x	a vector containing non-negative elements
parameter	parameter of the concentration measure (if set to NULL the default parameter of the respective measure is used)
type	character string giving the measure used to compute concentration. must be one of the strings in the default argument (the first character is sufficient). defaults to "Herfindahl".

Details

conc is just a wrapper for the concentration measures of Herfindahl and Rosenbluth (Hall / Tiedemann / Rosenbluth). If parameter is set to NULL the default from the respective function is used.

Value

the value of the concentration measure

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

F A Cowell: Measurement of Inequality, 2000, in A B Atkinson / F Bourguignon (Eds): Handbook of Income Distribution, Amsterdam,

F A Cowell: Measuring Inequality, 1995 Prentice Hall/Harvester Wheatshef,

M Hall / N Tidemann: Measures of Concentration, 1967, JASA 62, 162-168.

See Also

ineq, pov

Examples

```
# generate vector (of sales)
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
# compute Herfindahl coefficient with parameter 1
conc(x)
# compute coefficient of Hall/Tiedemann/Rosenbluth
conc(x, type="Rosenbluth")
```

`ineq` *Inequality Measures*

Description

computes the inequality within a vector according to the specified inequality measure

Usage

```
ineq(x, parameter = NULL, type = c("Gini", "RS", "Atkinson", "Theil", "Kolm", "v
     "square.var", "entropy"))

Gini(x)
RS(x)
Atkinson(x, parameter = 0.5)
Theil(x, parameter = 0)
Kolm(x, parameter = 1)
var.coeff(x, square = FALSE)
entropy(x, parameter = 0.5)
```

Arguments

`x`	a vector containing at least non-negative elements
`parameter`	parameter of the inequality measure (if set to `NULL` the default parameter of the respective measure is used)
`type`	character string giving the measure used to compute inequality. must be one of the strings in the default argument (the first character is sufficient). defaults to "Gini".
`square`	logical. Argument of the function `var.coeff`, for details see below.

Details

`ineq` is just a wrapper for the inequality measures `Gini`, `RS`, `Atkinson`, `Theil`, `Kolm`,`var.coeff`, `entropy`. If parameter is set to `NULL` the default from the respective function is used.

`Gini` is the Gini coefficient, `RS` is the the Ricci-Schutz coefficient (also called Pietrat's measure), `Atkinson` gives Atkinsont's measure and `Kolm` computes Kolmt's measure.

If the parameter in `Theil` is 0 Theilt's entropy measure is computed, for every other value Theilt's second measure is computed.

`ineq(x, type="var")` and `var.coeff(x)` respectively compute the coefficient of variation, while `ineq(x,type="square.var")` and `var.coeff(x, square=TRUE)` compute the squared coefficient of variation.

`entropy` computes the generalized entropy, which is for parameter 1 equal to Theilt's entropy coefficient and for parameter 0 equal to the second measure of Theil.

Value

the value of the inequality measure

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

F A Cowell: Measurement of Inequality, 2000, in A B Atkinson / F Bourguignon (Eds): Handbook of Income Distribution, Amsterdam,

F A Cowell: Measuring Inequality, 1995 Prentice Hall/Harvester Wheatshef,

Marshall / Olkin: Inequalities: Theory of Majorization and Its Applications, New York 1979 (Academic Press).

See Also

conc, pov

Examples

```
# generate vector (of incomes)
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
# compute Gini coefficient
ineq(x)
# compute Atkinson coefficient with parameter=0.5
ineq(x, parameter=0.5, type="Atkinson")
```

---

major *Majorization*

Description

tests whether a vector x majorizes another vector y

Usage

```
major(x,y)
```

Arguments

x,y	vectors containing non-negative elements (with same length and same mean)

Details

even if x and y are comparable (i.e. have same length and same mean) it is possible that neither x majorizes y nor y majorizes x.

Value

logical. TRUE if x majorizes y (x >=(M) y), FALSE if not.

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

Marshall / Olkin: Inequalities: Theory of Majorization and Its Applications, New York 1979 (Academic Press)

See Also

`Lc`

Examples

```
# generate vectors (of incomes)
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
y <- c(841, 2063, 2445, 3438, 4437, 5401, 6392, 8304, 11304, 21961)
# test whether x majorizes y (TRUE, because y is result of
# Pigou-Dalton-transfers)
major(x,y)
```

`plot.Lc` *Plot Lorenz Curve*

Description

plotting method for objects of class `"Lc"` (Lorenz curves)

Usage

```
## S3 method for class 'Lc':
plot(x, general=FALSE, lwd=2, xlab="p", ylab="L(p)",
    main="Lorenz curve", las=1, ...)
```

Arguments

`x`	an object of class `"Lc"`
`general`	logical. If TRUE the generalized Lorenz curve will be plotted
`lwd, xlab, ylab, main, las, ...`	high-level `plot` function parameters.

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

B C Arnold: Majorization and the Lorenz Order: A Brief Introduction, 1987, Springer,

F A Cowell: Measurement of Inequality, 2000, in A B Atkinson / F Bourguignon (Eds): Handbook of Income Distribution, Amsterdam,

F A Cowell: Measuring Inequality, 1995 Prentice Hall/Harvester Wheatshef.

See Also

`Lc`, `Lc.mehran`, `plot.theorLc`

Examples

```
## Load and attach income (and metadata) set from Ilocos, Philippines
data(Ilocos)
attach(Ilocos)
## extract and rescale income for the provinces "Pangasinan" und "La Union"
income.p <- income[province=="Pangasinan"]/10000
income.u <- income[province=="La Union"]/10000
## compute the Lorenz curves
Lc.p <- Lc(income.p)
Lc.u <- Lc(income.u)
## plot both Lorenz curves
plot(Lc.p)
lines(Lc.u, col=2)
```

---

plot.theorLc *Plot Theoretical Lorenz Curves*

Description

Plotting method for objects of class `"theorLc"` (theoretical Lorenz curves)

Usage

```
## S3 method for class 'theorLc':
plot(x, parameter=NULL, xlab="p", ylab="L(p)", lwd=2, las=1, ...)
```

Arguments

`x`	an object of class `"theorLc"`
`parameter`	vector containing parameters of the distributions. If `x` was generated by the function `theorLc` the parameters are already fixed and have to be set to `NULL`.
`xlab, ylab, lwd, las, ...`	high-level `plot` function parameters.

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

C Dagum: Income Distribution Models, 1983, in: Johnson / Kotz (Eds): Encyclopedia of Statistical Sciences Vol.4, 27-34.

J B McDonald: Some generalized functions for the size distribution of income, 1984, Econometrica 52, 647-664.

See Also

`Lc`, `plot.Lc`

Examples

```
# income distribution of the USA in 1968 (in 10 classes)
# x vector of class means, n vector of class frequencies
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
n <- c(482, 825, 722, 690, 661, 760, 745, 2140, 1911, 1024)

# compute minimal Lorenz curve (= no inequality in each group)
Lc.min <- Lc(x, n=n)
# compute maximal Lorenz curve (limits of Mehran)
Lc.max <- Lc.mehran(x,n)
# plot both Lorenz curves in one plot
plot(Lc.min)
lines(Lc.max, col=4)

# add the theoretic Lorenz curve of a Lognormal-distribution with variance 0.78
lines(Lc.lognorm, parameter=0.78)
# add the theoretic Lorenz curve of a Dagum-distribution
lines(Lc.dagum, parameter=c(3.4,2.6))
```

---

pov *Poverty Measures*

Description

computes the poverty of an (income) vector according to the specified poverty measure

Usage

```
pov(x, k, parameter = NULL,type = c("Watts", "Sen", "Foster"))

Watts(x,k)
Sen(x,k)
Foster(x,k,parameter=1)
```

Arguments

`x`	a vector containing at least non-negative elements
`k`	a constant giving the absolute poverty boundary
`parameter`	parameter of the poverty measure (if set to `NULL` the default parameter of the respective measure is used)
`type`	character string giving the measure used to compute poverty coefficient must be one of the strings in the default argument (the first character is sufficient). defaults to "Watts".

Details

`pov` is just a wrapper for the poverty measures of `Watts`, `Sen` and `Foster` (Foster / Greer / Thorbecke). If parameter is set to `NULL` the default from the respective function is used.

`Foster` gives for parameter 1 the head count ratio and for parameter 2 the poverty gap ratio.

Value

the value of the poverty measure

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

Buhong Zheng: Aggregate Poverty Measures, 1997, Journal of Economic Surveys Vol.11 No.2, 123-162.

J E Foster: On Economic Poverty: A Survey Of Aggregate Measures, 1984, Advances in Econometrics Vol. 3, 215-251.

See Also

`ineq`, `conc`

Examples

```
# generate vectors (of incomes)
x <- c(541, 1463, 2445, 3438, 4437, 5401, 6392, 8304, 11904, 22261)
y <- c(841, 2063, 2445, 3438, 4437, 5401, 6392, 8304, 11304, 21961)
# compute Watts index with poverty boundary 2000
pov(x, 2000)
pov(y, 2000)
# compute head count ratio with poverty boundary 2000
pov(x, 2000, parameter=1, type="Foster")
pov(y, 2000, parameter=1, type="Foster")
```

---

theorLc *Theoretical Lorenz Curves*

Description

Theoretical Lorenz curves of income distributions

Usage

```
theorLc(type=c("Singh-Maddala","Dagum","lognorm","Pareto","exponential"), parame

Lc.dagum(p, parameter=c(2,2))
Lc.singh(p, parameter=c(2,2))
Lc.pareto(p, parameter=2)
Lc.lognorm(p, parameter=1)
Lc.exp(p)
```

Arguments

`type`	character string giving the income distribution. Must be one of the strings in the default argument (the first character is sufficient). Defaults to "Singh-Maddala".
`parameter`	vector containing parameter(s) of the distributions.
`p`	vector with elements from [0,1].

Details

`Lc.dagum`, `Lc.singh`, `Lc.pareto`, `Lc.lognorm`, `Lc.exp` are theoretical Lorenz curves of income distributions. They are functions of class `"theorLc"` with plot- and a lines- method, so that they can be added into an existing Lorenz curve plot.

`theorLc` returns a function of class `"theorLc"`, that is a one of the above theoretical Lorenz curves with fixed parameters.

`Lc.dagum` is the Lorenz curve of the Dagum distribution (2 parameters), `Lc.singh` the one of the Singh-Maddala distribution (2 parameters), `Lc.pareto` the one of the Pareto distribution (1 parameter), `Lc.lognorm` the one of the Lognormal distribution (1 parameter) and `Lc.exp` the Lorenz curve of the exponential distribution (no parameter).

Value

A function of class `"theorLc"` or its value at `p` respectively.

Author(s)

Achim Zeileis ⟨zeileis@ci.tuwien.ac.at⟩

References

C Dagum: Income Distribution Models, 1983, in: Johnson / Kotz (Eds): Encyclopedia of Statistical Sciences Vol.4, 27-34.

J B McDonald: Some generalized functions for the size distribution of income, 1984, Econometrica 52, 647-664.

See Also

`Lc`, `plot.Lc`, `plot.theorLc`

Examples

```
## Load and attach income (and metadata) set from Ilocos, Philippines
data(Ilocos)
attach(Ilocos)
## extract income for the province "Pangasinan"
income.p <- income[province=="Pangasinan"]

## plot empirical Lorenz curve and add theoretical Lorenz curve of
## a lognormal distribution with an estimate of the standard
## deviation parameter
Lc.p <- Lc(income.p)
plot(Lc.p)
lines(Lc.lognorm, parameter=sd(log(income.p)), col=4)


# vector of percentages
p <- (1:10)*0.1
# compute values of theoretic Lorenz curve of a Dagum-distribution
Lc.dagum(p, parameter=c(3.4,2.6))
# or
mydagum <- theorLc(type="Dagum", parameter=c(3.4,2.6))
mydagum(p)
```

Index