Package: laeken (via r-universe)

September 3, 2024

Type Package

Title Estimation of Indicators on Social Exclusion and Poverty

Version 0.5.4

Date 2024-02-05

Depends R (>= 3.2.0)

Imports boot, MASS

Description Estimation of indicators on social exclusion and poverty,
as well as Pareto tail modeling for empirical income
distributions.

License GPL (>= 2)

Author Andreas Alfons [aut, cre]
(<https://orcid.org/0000-0002-2513-3788>), Josef Holzer [aut],
Matthias Templ [aut], Alexander Haider [ctb]

Maintainer Andreas Alfons <alfons@ese.eur.nl>

URL https://github.com/aalfons/laeken

BugReports https://github.com/aalfons/laeken/issues

Encoding UTF-8

RoxygenNote 7.2.3

Repository https://aalfons.r-universe.dev

RemoteUrl https://github.com/aalfons/laeken

RemoteRef HEAD

RemoteSha b96407d7e2e10c0db2ae45d94cae75e81a401a42

Contents

`laeken-package` *Estimation of Indicators on Social Exclusion and Poverty*

Description

Estimation of indicators on social exclusion and poverty, as well as Pareto tail modeling for empirical income distributions.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Andreas Alfons [aut, cre] (<https://orcid.org/0000-0002-2513-3788>), Josef Holzer [aut], Matthias Templ [aut], Alexander Haider [ctb]

Maintainer: Andreas Alfons <alfons@ese.eur.nl>

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2013) Robust estimation of economic indicators from survey samples based on Pareto tail modeling. *Journal of the Royal Statistical Society, Series C*, **62**(2), 271–286.

arpr *At-risk-of-poverty rate*

Description

Estimate the at-risk-of-poverty rate, which is defined as the proportion of persons with equivalized disposable income below the at-risk-of-poverty threshold.

Usage

```
arpr(
  inc,
  weights = NULL,
  sort = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  p = 0.6,
  var = NULL,
  alpha = 0.05,
  threshold = NULL,
  na.rm = FALSE,
  ...
)
```

Arguments

`inc` either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`weights` optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`sort` optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`breakdown` optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value. Note that the same (overall) threshold is used for all domains.

`design` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different strata for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`cluster` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`p` a numeric vector of values in $[0, 1]$ giving the percentages of the weighted median to be used for the at-risk-of-poverty threshold (see `arpt`).

`var` a character string specifying the type of variance estimation to be used, or `NULL` to omit variance estimation. See `variance` for possible values.

`alpha` numeric; if `var` is not `NULL`, this gives the significance level to be used for computing the confidence interval (i.e., the confidence level is $1-$`alpha`).

`threshold` if ‘NULL‘, the at-risk-at-poverty threshold is estimated from the data.

`na.rm` a logical indicating whether missing values should be removed.

`...` if `var` is not `NULL`, additional arguments to be passed to `variance`.

Details

The implementation strictly follows the Eurostat definition.

Value

A list of class `"arpr"` (which inherits from the class `"indicator"`) with the following components:

`value` a numeric vector containing the overall value(s).

`valueByStratum` a `data.frame` containing the values by domain, or `NULL`.

`varMethod` a character string specifying the type of variance estimation used, or `NULL` if variance estimation was omitted.

`var` a numeric vector containing the variance estimate(s), or `NULL`.

`varByStratum` a `data.frame` containing the variance estimates by domain, or `NULL`.

`ci` a numeric vector or matrix containing the lower and upper endpoints of the confidence interval(s), or `NULL`.

`ciByStratum` a `data.frame` containing the lower and upper endpoints of the confidence intervals by domain, or `NULL`.

`alpha` a numeric value giving the significance level used for computing the confidence interval(s) (i.e., the confidence level is $1-$`alpha`), or `NULL`.

`years` a numeric vector containing the different years of the survey.

`strata` a character vector containing the different domains of the breakdown.

`p` a numeric giving the percentage of the weighted median used for the at-risk-of-poverty threshold.

`threshold` a numeric vector containing the at-risk-of-poverty threshold(s).

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat, Luxembourg.

See Also

`arpt`, `variance`

Examples

```
data(eusilc)

# overall value
arpr("eqIncome", weights = "rb050", data = eusilc)

# values by region
arpr("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
```

arpt *At-risk-of-poverty threshold*

Description

Estimate the at-risk-of-poverty threshold. The standard definition is to use 60% of the weighted median equivalized disposable income.

Usage

```
arpt(
  inc,
  weights = NULL,
  sort = NULL,
  years = NULL,
  data = NULL,
  p = 0.6,
  na.rm = FALSE
)
```

Arguments

`inc`	either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`weights`	optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`sort`	optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`years`	optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.
`data`	an optional `data.frame`.
`p`	a numeric vector of values in $[0, 1]$ giving the percentages of the weighted median to be used for the at-risk-of-poverty threshold.
`na.rm`	a logical indicating whether missing values should be removed.

Details

The implementation strictly follows the Eurostat definition.

Value

A numeric vector containing the value(s) of the at-risk-of-poverty threshold is returned.

Author(s)

Andreas Alfons

References

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat.

See Also

`arpr`, `incMedian`, `weightedMedian`

Examples

```
data(eusilc)
arpt("eqIncome", weights = "rb050", data = eusilc)
```

`bootVar` *Bootstrap variance and confidence intervals of indicators on social exclusion and poverty*

Description

Compute variance and confidence interval estimates of indicators on social exclusion and poverty based on bootstrap resampling.

Usage

```
bootVar(
  inc,
  weights = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  indicator,
  R = 100,
  bootType = c("calibrate", "naive"),
  X,
  totals = NULL,
  ciType = c("perc", "norm", "basic"),
  alpha = 0.05,
  seed = NULL,
  na.rm = FALSE,
  gender = NULL,
  method = NULL,
  ...
)
```

Arguments

`inc` either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`weights` optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`breakdown` optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value.

`design` optional; either an integer vector or factor giving different strata for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, this is used as `strata` argument in the call to `boot`.

`cluster` optional; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`indicator` an object inheriting from the class `"indicator"` that contains the point estimates of the indicator (see `arpr`, `qsr`, `rmpg` or `gini`).

`R` a numeric value giving the number of bootstrap replicates.

`bootType` a character string specifying the type of bootstap to be performed. Possible values are `"calibrate"` (for calibration of the sample weights of the resampled observations in every iteration) and `"naive"` (for a naive bootstrap without calibration of the sample weights).

`X` if `bootType` is `"calibrate"`, a matrix of calibration variables.

`totals` numeric; if `bootType` is `"calibrate"`, this gives the population totals. If `years` is `NULL`, a vector should be supplied, otherwise a matrix in which each row contains the population totals of the respective year. If this is `NULL` (the default), the population totals are computed from the sample weights using the Horvitz-Thompson estimator.

`ciType` a character string specifying the type of confidence interval(s) to be computed. Possible values are `"perc"`, `"norm"` and `"basic"` (see `boot.ci`).

`alpha` a numeric value giving the significance level to be used for computing the confidence interval(s) (i.e., the confidence level is $1-$`alpha`), or `NULL`.

`seed` optional; an integer value to be used as the seed of the random number generator, or an integer vector containing the state of the random number generator to be restored.

`na.rm` a logical indicating whether missing values should be removed.

`gender` either a numeric vector giving the gender, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`method` a character string specifying the method to be used (only for `gpg`). Possible values are `"mean"` for the mean, and `"median"` for the median. If weights are provided, the weighted mean or weighted median is estimated.

`...` if `bootType` is `"calibrate"`, additional arguments to be passed to `calibWeights`.

Value

An object of the same class as `indicator` is returned. See `arpr`, `qsr`, `rmpg` or `gini` for details on the components.

Note

This function gives reasonable variance estimates for basic sample designs such as simple random sampling or stratified simple random sampling.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

See Also

`variance`, `calibWeights`, `arpr`, `qsr`, `rmpg`, `gini`

Examples

```
data(eusilc)
a <- arpr("eqIncome", weights = "rb050", data = eusilc)

## naive bootstrap
bootVar("eqIncome", weights = "rb050", design = "db040",
    data = eusilc, indicator = a, R = 50,
    bootType = "naive", seed = 123)

## bootstrap with calibration
bootVar("eqIncome", weights = "rb050", design = "db040",
    data = eusilc, indicator = a, R = 50,
    X = calibVars(eusilc$db040), seed = 123)
```

calibVars *Construct a matrix of binary variables for calibration*

Description

Construct a matrix of binary variables for calibration of sample weights according to known marginal population totals.

Usage

```
calibVars(x)
```

Arguments

x	a vector that can be interpreted as factor, or a matrix or `data.frame` consisting of such variables.

Value

A matrix of binary variables that indicate membership to the corresponding factor levels.

Author(s)

Andreas Alfons

See Also

`calibWeights`

Examples

```
data(eusilc)
# default method
aux <- calibVars(eusilc$rb090)
head(aux)
# data.frame method
aux <- calibVars(eusilc[, c("db040", "rb090")])
head(aux)
```

calibWeights *Calibrate sample weights*

Description

Calibrate sample weights according to known marginal population totals. Based on initial sample weights, the so-called *g*-weights are computed by generalized raking procedures.

Usage

```
calibWeights(
  X,
  d,
  totals,
  q = NULL,
  method = c("raking", "linear", "logit"),
  bounds = c(0, 10),
  maxit = 500,
  tol = 1e-06,
  eps = .Machine$double.eps
)
```

Arguments

`X`	a matrix of binary calibration variables (see `calibVars`).
`d`	a numeric vector giving the initial sample weights.
`totals`	a numeric vector of population totals corresponding to the calibration variables in `X`.
`q`	a numeric vector of positive values accounting for heteroscedasticity. Small values reduce the variation of the *g*-weights.
`method`	a character string specifying the calibration method to be used. Possible values are `"linear"` for the linear method, `"raking"` for the multiplicative method known as raking and `"logit"` for the logit method.
`bounds`	a numeric vector of length two giving bounds for the g-weights to be used in the logit method. The first value gives the lower bound (which must be smaller than or equal to 1) and the second value gives the upper bound (which must be larger than or equal to 1).
`maxit`	a numeric value giving the maximum number of iterations.
`tol`	the desired accuracy for the iterative procedure.
`eps`	the desired accuracy for computing the Moore-Penrose generalized inverse (see `ginv`).

Details

The final sample weights need to be computed by multiplying the resulting *g*-weights with the initial sample weights.

Value

A numeric vector containing the *g*-weights.

Note

This is a faster implementation of parts of `calib` from package `sampling`. Note that the default calibration method is raking and that the truncated linear method is not yet implemented.

Author(s)

Andreas Alfons

References

Deville, J.-C. and Särndal, C.-E. (1992) Calibration estimators in survey sampling. *Journal of the American Statistical Association*, **87**(418), 376–382.

Deville, J.-C., Särndal, C.-E. and Sautory, O. (1993) Generalized raking procedures in survey sampling. *Journal of the American Statistical Association*, **88**(423), 1013–1020.

See Also

`calibVars`, `bootVar`

Examples

```
data(eusilc)
# construct auxiliary 0/1 variables for genders
aux <- calibVars(eusilc$rb090)
# population totals
totals <- c(3990798, 4191431)
# compute g-weights
g <- calibWeights(aux, eusilc$rb050, totals)
# compute final weights
weights <- g * eusilc$rb050
summary(weights)
```

`eqInc` *Equivalized disposable income*

Description

Compute the equivalized disposable income from household and personal income variables.

Usage

```
eqInc(hid, hplus, hminus, pplus, pminus, eqSS, year = NULL, data = NULL)
```

Arguments

`hid` if `data=NULL`, a vector containing the household ID. Otherwise a character string specifying the column of `data` that contains the household ID.

`hplus` if `data=NULL`, a `data.frame` containing the household income components that have to be added. Otherwise a character vector specifying the columns of `data` that contain these income components.

`hminus` if `data=NULL`, a `data.frame` containing the household income components that have to be subtracted. Otherwise a character vector specifying the columns of `data` that contain these income components.

`pplus` if `data=NULL`, a `data.frame` containing the personal income components that have to be added. Otherwise a character vector specifying the columns of `data` that contain these income components.

`pminus` if `data=NULL`, a `data.frame` containing the personal income components that have to be subtracted. Otherwise a character vector specifying the columns of `data` that contain these income components.

`eqSS` if `data=NULL`, a vector containing the equivalized household size. Otherwise a character string specifying the column of `data` that contains the equivalized household size. See `eqSS` for more details.

`year` if `data=NULL`, a vector containing the year of the survey. Otherwise a character string specifying the column of `data` that contains the year.

`data` a `data.frame` containing EU-SILC survey data, or `NULL`.

Details

All income components should already be imputed, otherwise `NA`s are simply removed before the calculations.

Value

A numeric vector containing the equivalized disposable income for every individual in `data`.

Author(s)

Andreas Alfons

References

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat.

See Also

`eqSS`

Examples

```
data(eusilc)

# compute a simplified version of the equivalized disposable income
# (not all income components are available in the synthetic data)
hplus <- c("hy040n", "hy050n", "hy070n", "hy080n", "hy090n", "hy110n")
hminus <- c("hy130n", "hy145n")
pplus <- c("py010n", "py050n", "py090n", "py100n",
    "py110n", "py120n", "py130n", "py140n")
eqIncome <- eqInc("db030", hplus, hminus,
    pplus, character(), "eqSS", data=eusilc)

# combine with household ID and equivalized household size
tmp <- cbind(eusilc[, c("db030", "eqSS")], eqIncome)

# show the first 8 rows
head(tmp, 8)
```

eqSS *Equivalized household size*

Description

Compute the equivalized household size according to the modified OECD scale adopted in 1994.

Usage

```
eqSS(hid, age, year = NULL, data = NULL)
```

Arguments

`hid` if `data=NULL`, a vector containing the household ID. Otherwise a character string specifying the column of `data` that contains the household ID.

`age` if `data=NULL`, a vector containing the age of the individuals. Otherwise a character string specifying the column of `data` that contains the age.

`year` if `data=NULL`, a vector containing the year of the survey. Otherwise a character string specifying the column of `data` that contains the year.

`data` a `data.frame` containing EU-SILC survey data, or `NULL`.

Value

A numeric vector containing the equivalized household size for every observation in `data`.

Author(s)

Andreas Alfons

References

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat.

See Also

`eqInc`

Examples

```
data(eusilc)

# calculate equivalized household size
eqSS <- eqSS("db030", "age", data=eusilc)

# combine with household ID and household size
tmp <- cbind(eusilc[, c("db030", "hsize")], eqSS)

# show the first 8 rows
head(tmp, 8)
```

`eusilc` *Synthetic EU-SILC survey data*

Description

This data set is synthetically generated from real Austrian EU-SILC (European Union Statistics on Income and Living Conditions) data.

Usage

```
data(eusilc)
```

Format

A data frame with 14827 observations on the following 28 variables.

`db030` integer; the household ID.

`hsize` integer; the number of persons in the household.

`db040` factor; the federal state in which the household is located (levels `Burgenland`, `Carinthia`, `Lower Austria`, `Salzburg`, `Styria`, `Tyrol`, `Upper Austria`, `Vienna` and `Vorarlberg`).

`rb030` integer; the personal ID.

`age` integer; the person's age.

`rb090` factor; the person's gender (levels `male` and `female`).

pl030 factor; the person's economic status (levels 1 = working full time, 2 = working part time, 3 = unemployed, 4 = pupil, student, further training or unpaid work experience or in compulsory military or community service, 5 = in retirement or early retirement or has given up business, 6 = permanently disabled or/and unfit to work or other inactive person, 7 = fulfilling domestic tasks and care responsibilities).

pb220a factor; the person's citizenship (levels AT, EU and Other).

py010n numeric; employee cash or near cash income (net).

py050n numeric; cash benefits or losses from self-employment (net).

py090n numeric; unemployment benefits (net).

py100n numeric; old-age benefits (net).

py110n numeric; survivor's benefits (net).

py120n numeric; sickness benefits (net).

py130n numeric; disability benefits (net).

py140n numeric; education-related allowances (net).

hy040n numeric; income from rental of a property or land (net).

hy050n numeric; family/children related allowances (net).

hy070n numeric; housing allowances (net).

hy080n numeric; regular inter-household cash transfer received (net).

hy090n numeric; interest, dividends, profit from capital investments in unincorporated business (net).

hy110n numeric; income received by people aged under 16 (net).

hy130n numeric; regular inter-household cash transfer paid (net).

hy145n numeric; repayments/receipts for tax adjustment (net).

eqSS numeric; the equivalized household size according to the modified OECD scale.

eqIncome numeric; a slightly simplified version of the equivalized household income.

db090 numeric; the household sample weights.

rb050 numeric; the personal sample weights.

Details

The data set consists of 6000 households and is used in the examples of package laeken. Note that this is a synthetic data set based on original EU-SILC survey data.

Only a few of the large number of variables in the original survey are included in this example data set. The variable names are rather cryptic codes, but these are the standardized names used by the statistical agencies. Furthermore, the variables hsize, age, eqSS and eqIncome are not included in the standardized format of EU-SILC data, but have been derived from other variables for convenience. Moreover, some very sparse income components were not included in the the generation of this synthetic data set. Thus the equivalized household income is computed from the available income components.

Source

This is a synthetic data set based on Austrian EU-SILC data from 2006. The original sample was provided by Statistics Austria.

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2011) Simulation of close-to-reality population data for household surveys with application to EU-SILC. *Statistical Methods and Applications*, vol 20 (3), 383-407.

Eurostat (2004) Description of target variables: Cross-sectional and longitudinal. *EU-SILC 065/04*, Eurostat.

Examples

```
data(eusilc)
summary(eusilc)
```

`fitPareto` *Fit income distribution models with the Pareto distribution*

Description

Fit a Pareto distribution to the upper tail of income data. Since a theoretical distribution is used for the upper tail, this is a semiparametric approach.

Usage

```
fitPareto(
  x,
  k = NULL,
  x0 = NULL,
  method = "thetaPDC",
  groups = NULL,
  w = NULL,
  ...
)
```

Arguments

x	a numeric vector.
k	the number of observations in the upper tail to which the Pareto distribution is fitted.
x0	the threshold (scale parameter) above which the Pareto distribution is fitted.
method	either a function or a character string specifying the function to be used to estimate the shape parameter of the Pareto distibution, such as `thetaPDC` (the default). See “Details” for requirements for such a function and “See also” for available functions.

groups an optional vector or factor specifying groups of elements of x (e.g., households). If supplied, each group of observations is expected to have the same value in x (e.g., household income). Only the values of every first group member to appear are used for fitting the Pareto distribution. For each group above the threshold, every group member is assigned the same value.

w an optional numeric vector giving sample weights.

... addtional arguments to be passed to the specified method.

Details

The arguments k and x0 of course correspond with each other. If k is supplied, the threshold x0 is estimated with the $n - k$ largest value in x, where n is the number of observations. On the other hand, if the threshold x0 is supplied, k is given by the number of observations in x larger than x0. Therefore, either k or x0 needs to be supplied. If both are supplied, only k is used (mainly for back compatibility).

The function supplied to method should take a numeric vector (the observations) as its first argument. If k is supplied, it will be passed on (in this case, the function is required to have an argument called k). Similarly, if the threshold x0 is supplied, it will be passed on (in this case, the function is required to have an argument called x0). As above, only k is passed on if both are supplied. If the function specified by method can handle sample weights, the corresponding argument should be called w. Additional arguments are passed via the ... argument.

Value

A numeric vector with a Pareto distribution fit to the upper tail.

Note

The arguments x0 for the threshold (scale parameter) of the Pareto distribution and w for sample weights were introduced in version 0.2. This results in slightly different behavior regarding the function calls to method compared to prior versions.

Author(s)

Andreas Alfons and Josef Holzer

See Also

paretoTail, replaceTail

thetaPDC, thetaWML, thetaHill, thetaISE, thetaLS, thetaMoment, thetaQQ, thetaTM

Examples

```
data(eusilc)


## gini coefficient without Pareto tail modeling
gini("eqIncome", weights = "rb050", data = eusilc)
```

```
## gini coefficient with Pareto tail modeling

# using number of observations in tail
eqIncome <- fitPareto(eusilc$eqIncome, k = 175,
    w = eusilc$db090, groups = eusilc$db030)
gini(eqIncome, weights = eusilc$rb050)

# using threshold
eqIncome <- fitPareto(eusilc$eqIncome, x0 = 44150,
    w = eusilc$db090, groups = eusilc$db030)
gini(eqIncome, weights = eusilc$rb050)
```

gini *Gini coefficient*

Description

Estimate the Gini coefficient, which is a measure for inequality.

Usage

```
gini(
  inc,
  weights = NULL,
  sort = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  var = NULL,
  alpha = 0.05,
  na.rm = FALSE,
  ...
)
```

Arguments

`inc`	either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`weights`	optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`sort` optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`breakdown` optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value.

`design` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different domains for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`cluster` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`var` a character string specifying the type of variance estimation to be used, or `NULL` to omit variance estimation. See `variance` for possible values.

`alpha` numeric; if `var` is not `NULL`, this gives the significance level to be used for computing the confidence interval (i.e., the confidence level is $1-$`alpha`).

`na.rm` a logical indicating whether missing values should be removed.

`...` if `var` is not `NULL`, additional arguments to be passed to `variance`.

Details

The implementation strictly follows the Eurostat definition.

Value

A list of class `"gini"` (which inherits from the class `"indicator"`) with the following components:

`value` a numeric vector containing the overall value(s).

`valueByStratum` a `data.frame` containing the values by domain, or `NULL`.

`varMethod` a character string specifying the type of variance estimation used, or `NULL` if variance estimation was omitted.

`var` a numeric vector containing the variance estimate(s), or `NULL`.

`varByStratum` a `data.frame` containing the variance estimates by domain, or `NULL`.

`ci` a numeric vector or matrix containing the lower and upper endpoints of the confidence interval(s), or `NULL`.

`ciByStratum` a `data.frame` containing the lower and upper endpoints of the confidence intervals by domain, or `NULL`.

`alpha` a numeric value giving the significance level used for computing the confidence interval(s) (i.e., the confidence level is 1−`alpha`), or `NULL`.

`years` a numeric vector containing the different years of the survey.

`strata` a character vector containing the different domains of the breakdown.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat, Luxembourg.

See Also

`variance`, `qsr`

Examples

```
data(eusilc)

# overall value
gini("eqIncome", weights = "rb050", data = eusilc)

# values by region
gini("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
```

gpg *Gender pay (wage) gap.*

Description

Estimate the gender pay (wage) gap.

Usage

```
gpg(
  inc,
  gender = NULL,
  method = c("mean", "median"),
  weights = NULL,
  sort = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  var = NULL,
  alpha = 0.05,
  na.rm = FALSE,
  ...
)
```

Arguments

`inc` either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`gender` either a factor giving the gender, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`method` a character string specifying the method to be used. Possible values are `"mean"` for the mean, and `"median"` for the median. If weights are provided, the weighted mean or weighted median is estimated.

`weights` optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`sort` optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`breakdown` optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value.

`design` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different strata for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`cluster` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`var` a character string specifying the type of variance estimation to be used, or `NULL` to omit variance estimation. See `variance` for possible values.

`alpha` numeric; if `var` is not `NULL`, this gives the significance level to be used for computing the confidence interval (i.e., the confidence level is 1−`alpha`).

na.rm a logical indicating whether missing values should be removed.

... if var is not NULL, additional arguments to be passed to variance.

Details

The implementation strictly follows the Eurostat definition (with default method "mean" and alternative method "median"). If weights are provided, the weighted mean or weighted median is estimated.

Value

A list of class "gpg" (which inherits from the class "indicator") with the following components:

value a numeric vector containing the overall value(s).

valueByStratum a data.frame containing the values by domain, or NULL.

varMethod a character string specifying the type of variance estimation used, or NULL if variance estimation was omitted.

var a numeric vector containing the variance estimate(s), or NULL.

varByStratum a data.frame containing the variance estimates by domain, or NULL.

ci a numeric vector or matrix containing the lower and upper endpoints of the confidence interval(s), or NULL.

ciByStratum a data.frame containing the lower and upper endpoints of the confidence intervals by domain, or NULL.

alpha a numeric value giving the significance level used for computing the confidence interv al(s) (i.e., the confidence level is 1−alpha), or NULL.

years a numeric vector containing the different years of the survey.

strata a character vector containing the different domains of the breakdown.

Author(s)

Matthias Templ and Alexander Haider, using code for breaking down estimation by Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat, Luxembourg.

See Also

variance, qsr, gini

Examples

```
data(ses)

# overall value with mean
gpg("earningsHour", gender = "sex", weigths = "weights",
    data = ses)

# overall value with median
gpg("earningsHour", gender = "sex", weigths = "weights",
    data = ses, method = "median")

# values by education with mean
gpg("earningsHour", gender = "sex", weigths = "weights",
    breakdown = "education", data = ses)

# values by education with median
gpg("earningsHour", gender = "sex", weigths = "weights",
    breakdown = "education", data = ses, method = "median")
```

incMean *Weighted mean income*

Description

Compute the weighted mean income.

Usage

```
incMean(inc, weights = NULL, years = NULL, data = NULL, na.rm = FALSE)
```

Arguments

`inc`	either a numeric vector giving the (equivalized disposable) income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`weights`	optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`years`	optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.
`data`	an optional `data.frame`.
`na.rm`	a logical indicating whether missing values should be removed.

Value

A numeric vector containing the value(s) of the weighted mean income is returned.

Author(s)

Andreas Alfons

See Also

weightedMean

Examples

```
data(eusilc)
incMean("eqIncome", weights = "rb050", data = eusilc)
```

incMedian *Weighted median income*

Description

Compute the weighted median income.

Usage

```
incMedian(
  inc,
  weights = NULL,
  sort = NULL,
  years = NULL,
  data = NULL,
  na.rm = FALSE
)
```

Arguments

`inc` either a numeric vector giving the (equivalized disposable) income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`weights` optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`sort` optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`data` an optional `data.frame`.

`na.rm` a logical indicating whether missing values should be removed.

Details

The implementation strictly follows the Eurostat definition.

Value

A numeric vector containing the value(s) of the weighted median income is returned.

Author(s)

Andreas Alfons

References

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat.

See Also

`arpt`, `weightedMedian`

Examples

```
data(eusilc)
incMedian("eqIncome", weights = "rb050", data = eusilc)
```

`incQuintile` *Weighted income quintile*

Description

Compute weighted income quintiles.

Usage

```
incQuintile(
  inc,
  weights = NULL,
  sort = NULL,
  years = NULL,
  k = c(1, 4),
  data = NULL,
  na.rm = FALSE
)
```

Arguments

inc	either a numeric vector giving the (equivalized disposable) income, or (if data is not NULL) a character string, an integer or a logical vector specifying the corresponding column of data.
weights	optional; either a numeric vector giving the personal sample weights, or (if data is not NULL) a character string, an integer or a logical vector specifying the corresponding column of data.
sort	optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if data is not NULL) a character string, an integer or a logical vector specifying the corresponding column of data.
years	optional; either a numeric vector giving the different years of the survey, or (if data is not NULL) a character string, an integer or a logical vector specifying the corresponding column of data. If supplied, values are computed for each year.
k	a vector of integers between 0 and 5 specifying the quintiles to be computed (0 gives the minimum, 5 the maximum).
data	an optional data.frame.
na.rm	a logical indicating whether missing values should be removed.

Details

The implementation strictly follows the Eurostat definition.

Value

A numeric vector (if years is NULL) or matrix (if years is not NULL) containing the values of the weighted income quintiles specified by k are returned.

Author(s)

Andreas Alfons

References

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat.

See Also

qsr, weightedQuantile

Examples

```
data(eusilc)
incQuintile("eqIncome", weights = "rb050", data = eusilc)
```

meanExcessPlot *Mean excess plot*

Description

The Mean Excess plot is a graphical method for detecting the threshold (scale parameter) of a Pareto distribution.

Usage

```
meanExcessPlot(
  x,
  w = NULL,
  probs = NULL,
  interactive = TRUE,
  pch = par(”pch”),
  cex = par(”cex”),
  col = par(”col”),
  bg = ”transparent”,
  ...
)
```

Arguments

`x` a numeric vector.

`w` an optional numeric vector giving sample weights.

`probs` an optional numeric vector of probabilities with values in $[0, 1]$, defining the quantiles to be plotted. This is useful for large data sets, when it may not be desirable to plot every single point.

`interactive` a logical indicating whether the threshold (scale parameter) can be selected interactively by clicking on points. Information on the selected threshold is then printed on the console.

`pch, cex, col, bg` graphical parameters for the plot symbol of each data point or quantile (see `points`).

`...` additional arguments to be passed to `plot.default`.

Details

The corresponding mean excesses are plotted against the values of `x` (if supplied, only those specified by `probs`). If the tail of the data follows a Pareto distribution, these observations show a positive linear trend. The leftmost point of a fitted line can thus be used as an estimate of the threshold (scale parameter).

The interactive selection of the threshold (scale parameter) is implemented using `identify`. For the usual `X11` device, the selection process is thus terminated by pressing any mouse button other than the first. For the `quartz` device (on Mac OS X systems), the process is terminated either by a secondary click (usually second mouse button or `Ctrl`-click) or by pressing the `ESC` key.

Value

If interactive is TRUE, the last selection for the threshold is returned invisibly as an object of class "paretoScale", which consists of the following components:

x0	the selected threshold (scale parameter).
k	the number of observations in the tail (i.e., larger than the threshold).

Note

The functionality to account for sample weights and to select the threshold (scale parameter) interactively was introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

See Also

paretoScale, paretoTail, minAMSE, paretoQPlot, identify

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# with sample weights
meanExcessPlot(eusilc$eqIncome, w = eusilc$db090)

# without sample weights
meanExcessPlot(eusilc$eqIncome)
```

minAMSE *Weighted asymptotic mean squared error (AMSE) estimator*

Description

Estimate the scale and shape parameters of a Pareto distribution with an iterative procedure based on minimizing the weighted asymptotic mean squared error (AMSE) of the Hill estimator.

Usage

```
minAMSE(
  x,
  weight = c("Bernoulli", "JASA"),
  kmin,
  kmax,
  mmax,
  tol = 0,
  maxit = 100
)

## S3 method for class 'minAMSE'
print(x, ...)
```

Arguments

`x` for `minAMSE`, a numeric vector. The `print` method is called by the generic function if an object of class `"minAMSE"` is supplied.

`weight` a character vector specifying the weighting scheme to be used in the procedure. If `"Bernoulli"`, the weight functions as described in the *Bernoulli* paper are applied. If `"JASA"`, the weight functions as described in the *Journal of the Americal Statistical Association* are used.

`kmin` An optional integer giving the lower bound for finding the optimal number of observations in the tail. It defaults to $[\frac{n}{100}]$, where n denotes the number of observations in `x` (see the references).

`kmax` An optional integer giving the upper bound for finding the optimal number of observations in the tail (see “Details”).

`mmax` An optional integer giving the upper bound for finding the optimal number of observations for computing the nuisance parameter ρ (see “Details” and the references).

`tol` an integer giving the desired tolerance level for finding the optimal number of observations in the tail.

`maxit` a positive integer giving the maximum number of iterations.

`...` additional arguments to be passed to `print.default`.

Details

The weights used in the weighted AMSE depend on a nuisance parameter ρ. Both the optimal number of observations in the tail and the nuisance parameter ρ are estimated iteratively using nonlinear integer minimization. This is currently done by a brute force algorithm, hence it is stronly recommended to supply upper bounds `kmax` and `mmax`.

See the references for more details on the iterative algorithm.

Value

An object of class `"minAMSE"` with the following components:

`kopt` the optimal number of observations in the tail.

`x0` the corresponding threshold.

`theta` the estimated shape parameter of the Pareto distribution.

`MSEmin` the minimal MSE.

`rho` the estimated nuisance parameter.

`k` the examined range for the number of observations in the tail.

`MSE` the corresponding MSEs.

Author(s)

Josef Holzer and Andreas Alfons

References

Beirlant, J., Vynckier, P. and Teugels, J.L. (1996) Tail index estimation, Pareto quantile plots, and regression diagnostics. *Journal of the American Statistical Association*, **91**(436), 1659–1667.

Beirlant, J., Vynckier, P. and Teugels, J.L. (1996) Excess functions and estimation of the extreme-value index. *Bernoulli*, **2**(4), 293–318.

Dupuis, D.J. and Victoria-Feser, M.-P. (2006) A robust prediction error criterion for Pareto modelling of upper tails. *The Canadian Journal of Statistics*, **34**(4), 639–658.

See Also

`thetaHill`

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
minAMSE(eusilc$eqIncome[!duplicated(eusilc$db030)],
    kmin = 60, kmax = 150, mmax = 250)
```

`paretoQPlot` *Pareto quantile plot*

Description

The Pareto quantile plot is a graphical method for inspecting the parameters of a Pareto distribution.

Usage

```
paretoQPlot(
  x,
  w = NULL,
  xlab = NULL,
  ylab = NULL,
  interactive = TRUE,
  x0 = NULL,
  theta = NULL,
  pch = par("pch"),
  cex = par("cex"),
  col = par("col"),
  bg = "transparent",
  ...
)
```

Arguments

`x` a numeric vector.

`w` an optional numeric vector giving sample weights.

`xlab`, `ylab` axis labels.

`interactive` a logical indicating whether the threshold (scale parameter) can be selected interactively by clicking on points. Information on the selected threshold is then printed on the console.

`x0`, `theta` optional; if estimates of the threshold (scale parameter) and the shape parameter have already been obtained, they can be passed through the corresponding argument (`x0` for the threshold, `theta` for the shape parameter). If both arguments are supplied and `interactive` is not `TRUE`, reference lines are drawn to indicate the parameter estimates.

`pch`, `cex`, `col`, `bg` graphical parameters for the plot symbol of each data point (see `points`).

`...` additional arguments to be passed to `plot.default`.

Details

If the Pareto model holds, there exists a linear relationship between the lograrithms of the observed values and the quantiles of the standard exponential distribution, since the logarithm of a Pareto distributed random variable follows an exponential distribution. Hence the logarithms of the observed values are plotted against the corresponding theoretical quantiles. If the tail of the data follows a Pareto distribution, these observations form almost a straight line. The leftmost point of a fitted line can thus be used as an estimate of the threshold (scale parameter). The slope of the fitted line is in turn an estimate of $\frac{1}{\theta}$, the reciprocal of the shape parameter.

The interactive selection of the threshold (scale parameter) is implemented using `identify`. For the usual `X11` device, the selection process is thus terminated by pressing any mouse button other than the first. For the `quartz` device (on Mac OS X systems), the process is terminated either by a secondary click (usually second mouse button or `Ctrl`-click) or by pressing the `ESC` key.

Value

If `interactive` is `TRUE`, the last selection for the threshold is returned invisibly as an object of class `"paretoScale"`, which consists of the following components:

`x0`	the selected threshold (scale parameter).
`k`	the number of observations in the tail (i.e., larger than the threshold).

Note

The functionality to account for sample weights and to select the threshold (scale parameter) interactively was introduced in version 0.2. Also starting with version 0.2, a logarithmic y-axis is now used to display the axis labels in the scale of the original values.

Author(s)

Andreas Alfons and Josef Holzer

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2013) Robust estimation of economic indicators from survey samples based on Pareto tail modeling. *Journal of the Royal Statistical Society, Series C*, **62**(2), 271–286.

Beirlant, J., Vynckier, P. and Teugels, J.L. (1996) Tail index estimation, Pareto quantile plots, and regression diagnostics. *Journal of the American Statistical Association*, **91**(436), 1659–1667.

See Also

`paretoScale`, `paretoTail`, `minAMSE`, `meanExcessPlot`, `identify`

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# with sample weights
paretoQPlot(eusilc$eqIncome, w = eusilc$db090)

# without sample weights
paretoQPlot(eusilc$eqIncome)
```

paretoScale *Estimate the scale parameter of a Pareto distribution*

Description

Estimate the scale parameter of a Pareto distribution, i.e., the threshold for Pareto tail modeling.

Usage

```
paretoScale(
  x,
  w = NULL,
  groups = NULL,
  method = "VanKerm",
  center = c("mean", "median"),
  probs = c(0.97, 0.98),
  na.rm = FALSE
)
```

Arguments

x	a numeric vector.
w	an optional numeric vector giving sample weights.
groups	an optional vector or factor specifying groups of elements of x (e.g., households). If supplied, each group of observations is expected to have the same value in x (e.g., household income). Only the values of every first group member to appear are used for estimating the threshold (scale parameter).
method	a character string specifying the estimation method. If "VanKerm", Van Kerm's method is used, which is a rule of thumb specifically designed for the equivalized disposable income in EU-SILC data (currently the only method implemented).
center	a character string specifying the estimation method for the center of the distribution. Possible values are "mean" for the weighted mean and "median" for the weighted median. This is used if method is "VanKerm" (currently the only method implemented).
probs	a numeric vector of length two giving probabilities to be used for computing weighted quantiles of the distribution. Values should be close to 1 such that the quantiles correspond to the upper tail. This is used if method is "VanKerm" (currently the only method implemented).
na.rm	a logical indicating whether missing values in x should be omitted.

Details

Van Kerm's formula is given by

$$\min(\max(2.5\bar{x}, q(0.98), q(0.97))),$$

where $\bar{x}$ denotes the weighted mean and $q(.)$ denotes weighted quantiles. This function allows to compute generalizations of Van Kerm's formula, where the mean can be replaced by the median and different quantiles can be used.

Value

An object of class "paretoScale" with the following components:

x0	the threshold (scale parameter).
k	the number of observations in the tail (i.e., larger than the threshold).

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

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

See Also

minAMSE, paretoQPlot, meanExcessPlot

Examples

```
data(eusilc)
paretoScale(eusilc$eqIncome, eusilc$db090, groups = eusilc$db030)
```

paretoTail *Pareto tail modeling for income distributions*

Description

Fit a Pareto distribution to the upper tail of income data. Since a theoretical distribution is used for the upper tail, this is a semiparametric approach.

Usage

```
paretoTail(
  x,
  k = NULL,
  x0 = NULL,
  method = "thetaPDC",
  groups = NULL,
  w = NULL,
  alpha = 0.01,
  ...
)
```

Arguments

`x` a numeric vector.

`k` the number of observations in the upper tail to which the Pareto distribution is fitted.

`x0` the threshold (scale parameter) above which the Pareto distribution is fitted.

`method` either a function or a character string specifying the function to be used to estimate the shape parameter of the Pareto distibution, such as `thetaPDC` (the default). See “Details” for requirements for such a function and “See also” for available functions.

`groups` an optional vector or factor specifying groups of elements of `x` (e.g., households). If supplied, each group of observations is expected to have the same value in `x` (e.g., household income). Only the values of every first group member to appear are used for fitting the Pareto distribution.

`w` an optional numeric vector giving sample weights.

`alpha` numeric; values above the theoretical 1−`alpha` quantile of the fitted Pareto distribution will be flagged as outliers for further treatment with `reweightOut` or `replaceOut`.

`...` addtional arguments to be passed to the specified method.

Details

The arguments `k` and `x0` of course correspond with each other. If `k` is supplied, the threshold `x0` is estimated with the $n - k$ largest value in `x`, where n is the number of observations. On the other hand, if the threshold `x0` is supplied, `k` is given by the number of observations in `x` larger than `x0`. Therefore, either `k` or `x0` needs to be supplied. If both are supplied, only `k` is used.

The function supplied to `method` should take a numeric vector (the observations) as its first argument. If `k` is supplied, it will be passed on (in this case, the function is required to have an argument called `k`). Similarly, if the threshold `x0` is supplied, it will be passed on (in this case, the function is required to have an argument called `x0`). As above, only `k` is passed on if both are supplied. If the function specified by `method` can handle sample weights, the corresponding argument should be called `w`. Additional arguments are passed via the ... argument.

Value

An object of class `"paretoTail"` with the following components:

`x` the supplied numeric vector.

`k` the number of observations in the upper tail to which the Pareto distribution has been fitted.

`groups` if supplied, the vector or factor specifying groups of elements.

`w` if supplied, the numeric vector of sample weights.

`method` the function used to estimate the shape parameter, or the name of the function.

`x0` the scale parameter.

`theta` the estimated shape parameter.

`tail` if `groups` is not `NULL`, this gives the groups with values larger than the threshold (scale parameter), otherwise the indices of observations in the upper tail.

`alpha` the tuning parameter `alpha` used for flagging outliers.

`out` if `groups` is not `NULL`, this gives the groups that are flagged as outliers, otherwise the indices of the flagged observations.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2013) Robust estimation of economic indicators from survey samples based on Pareto tail modeling. *Journal of the Royal Statistical Society, Series C*, **62**(2), 271–286.

See Also

`reweightOut`, `shrinkOut`, `replaceOut`, `replaceTail`, `fitPareto`

`thetaPDC`, `thetaWML`, `thetaHill`, `thetaISE`, `thetaLS`, `thetaMoment`, `thetaQQ`, `thetaTM`

Examples

```
data(eusilc)


## gini coefficient without Pareto tail modeling
gini("eqIncome", weights = "rb050", data = eusilc)


## gini coefficient with Pareto tail modeling

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090,
    groups = eusilc$db030)

# estimate shape parameter
fit <- paretoTail(eusilc$eqIncome, k = ts$k,
    w = eusilc$db090, groups = eusilc$db030)

# calibration of outliers
w <- reweightOut(fit, calibVars(eusilc$db040))
```

```
gini(eusilc$eqIncome, w)

# winsorization of outliers
eqIncome <- shrinkOut(fit)
gini(eqIncome, weights = eusilc$rb050)

# replacement of outliers
eqIncome <- replaceOut(fit)
gini(eqIncome, weights = eusilc$rb050)

# replacement of whole tail
eqIncome <- replaceTail(fit)
gini(eqIncome, weights = eusilc$rb050)
```

plot.paretoTail *Diagnostic plot for the Pareto tail model*

Description

Produce a diagnostic Pareto quantile plot for evaluating the fitted Pareto distribution. Reference lines indicating the estimates of the threshold (scale parameter) and the shape parameter are added to the plot, and any detected outliers are highlighted.

Usage

```
## S3 method for class 'paretoTail'
plot(
  x,
  pch = c(1, 3),
  cex = 1,
  col = c("black", "red"),
  bg = "transparent",
  ...
)
```

Arguments

x	an object of class "paretoTail" as returned by paretoTail.
pch, cex, col, bg	graphical parameters. Each can be a vector of length two, with the first and second element giving the graphical parameter for the good data points and the outliers, respectively.
...	additional arguments to be passed to paretoQPlot.

Details

While the first horizontal line indicates the estimated threshold (scale parameter), the estimated shape parameter is indicated by a line whose slope is given by the reciprocal of the estimate. In addition, the second horizontal line represents the theoretical quantile of the fitted distribution that is used for outlier detection. Thus all values above that line are the detected outliers.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

See Also

paretoTail, paretoQPlot

Examples

```
data(eusilc)

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090,
    groups = eusilc$db030)

# estimate shape parameter
fit <- paretoTail(eusilc$eqIncome, k = ts$k,
    w = eusilc$db090, groups = eusilc$db030)

# produce plot
plot(fit)
```

prop *Proportion of an alternative distribution*

Description

Estimate the proportion of an alternative distribution.

Usage

```
prop(
  bin,
  weights = NULL,
  sort = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  var = NULL,
  alpha = 0.05,
  na.rm = FALSE,
  ...
)
```

Arguments

`bin` either a factor vector giving the values, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`weights` optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`sort` optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`breakdown` optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value.

`design` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different domains for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`cluster` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`var` a character string specifying the type of variance estimation to be used, or `NULL` to omit variance estimation. See `variance` for possible values.

`alpha` numeric; if `var` is not `NULL`, this gives the significance level to be used for computing the confidence interval (i.e., the confidence level is $1-$`alpha`).

`na.rm` a logical indicating whether missing values should be removed.

`...` if `var` is not `NULL`, additional arguments to be passed to `variance`.

Details

If weights are provided, the weighted proportion is estimated.

Value

A list of class `"prop"` (which inherits from the class `"indicator"`) with the following components:

`value` a numeric vector containing the overall value(s).

`valueByStratum` a `data.frame` containing the values by domain, or `NULL`.

`varMethod` a character string specifying the type of variance estimation used, or `NULL` if variance estimation was omitted.

`var` a numeric vector containing the variance estimate(s), or `NULL`.

`varByStratum` a `data.frame` containing the variance estimates by domain, or `NULL`.

`ci` a numeric vector or matrix containing the lower and upper endpoints of the confidence interval(s), or `NULL`.

`ciByStratum` a `data.frame` containing the lower and upper endpoints of the confidence intervals by domain, or `NULL`.

`alpha` a numeric value giving the significance level used for computing the confidence interval(s) (i.e., the confidence level is $1-$`alpha`), or `NULL`.

`years` a numeric vector containing the different years of the survey.

`strata` a character vector containing the different domains of the breakdown.

Author(s)

Matthias Templ, using code for breaking down estimation by Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat, Luxembourg.

See Also

`variance`

Examples

```
data(eusilc)

# overall value
prop("rb090", weights = "rb050", data = eusilc)

# values by region
p1 <- prop("rb090", weights = "rb050",
    breakdown = "db040",  cluster = "db030",
    data = eusilc)

p1

## Not run:
variance("rb090", weights = "rb050",
    breakdown = "db040", data = eusilc, indicator=p1,
    cluster="db030", X = calibVars(eusilc$db040))
```

```
## End(Not run)


eusilc$agecut <- cut(eusilc$age, 2)
p1 <- prop("agecut", weights = "rb050",
           breakdown = "db040",
           cluster="db030", data = eusilc)
p1

## Not run:
variance("agecut", weights = "rb050",
         breakdown = "db040", data = eusilc, indicator=p1,
         X = calibVars(eusilc$db040), cluster="db030")

## End(Not run)


eusilc$eqIncomeCat <- factor(ifelse(eusilc$eqIncome < quantile(eusilc$eqIncome,0.2), "one", "two"))
p1 <- prop("eqIncomeCat", weights = "rb050",
           breakdown = "db040", data = eusilc, cluster="db030")
p1

## Not run:
variance("eqIncomeCat", weights = "rb050",
         breakdown = "db040", data = eusilc, indicator=p1,
         X = calibVars(eusilc$db040), cluster="db030")

## End(Not run)
```

qsr *Quintile share ratio*

Description

Estimate the quintile share ratio, which is defined as the ratio of the sum of equivalized disposable income received by the top 20% to the sum of equivalized disposable income received by the bottom 20%.

Usage

```
qsr(
  inc,
  weights = NULL,
  sort = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  var = NULL,
  alpha = 0.05,
  na.rm = FALSE,
  ...
)
```

Arguments

`inc` either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`weights` optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`sort` optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`breakdown` optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value.

`design` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different strata for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`cluster` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`var` a character string specifying the type of variance estimation to be used, or `NULL` to omit variance estimation. See `variance` for possible values.

`alpha` numeric; if `var` is not `NULL`, this gives the significance level to be used for computing the confidence interval (i.e., the confidence level is $1-$`alpha`).

`na.rm` a logical indicating whether missing values should be removed.

`...` if `var` is not `NULL`, additional arguments to be passed to `variance`.

Details

The implementation strictly follows the Eurostat definition.

Value

A list of class `"qsr"` (which inherits from the class `"indicator"`) with the following components:

`value` a numeric vector containing the overall value(s).

`valueByStratum` a `data.frame` containing the values by domain, or `NULL`.

`varMethod` a character string specifying the type of variance estimation used, or `NULL` if variance estimation was omitted.

`var` a numeric vector containing the variance estimate(s), or `NULL`.

`varByStratum` a `data.frame` containing the variance estimates by domain, or `NULL`.

`ci` a numeric vector or matrix containing the lower and upper endpoints of the confidence interval(s), or `NULL`.

`ciByStratum` a `data.frame` containing the lower and upper endpoints of the confidence intervals by domain, or `NULL`.

`alpha` a numeric value giving the significance level used for computing the confidence interval(s) (i.e., the confidence level is $1-$`alpha`), or `NULL`.

`years` a numeric vector containing the different years of the survey.

`strata` a character vector containing the different domains of the breakdown.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat, Luxembourg.

See Also

`incQuintile`, `variance`, `gini`

Examples

```
data(eusilc)

# overall value
qsr("eqIncome", weights = "rb050", data = eusilc)

# values by region
qsr("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
```

replaceTail *Replace observations under a Pareto model*

Description

Replace observations under a Pareto model for the upper tail with values drawn from the fitted distribution.

Usage

```
replaceTail(x, ...)

## S3 method for class 'paretoTail'
replaceTail(x, all = TRUE, ...)

replaceOut(x, ...)
```

Arguments

`x`	an object of class `"paretoTail"` (see `paretoTail`).
`...`	additional arguments to be passed down.
`all`	a logical indicating whether all observations in the upper tail should be replaced or only those flagged as outliers.

Details

`replaceOut(x, ...{})` is a simple wrapper for `replaceTail(x,all = FALSE, ...{})`.

Value

A numeric vector consisting mostly of the original values, but with observations in the upper tail replaced with values from the fitted Pareto distribution.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2013) Robust estimation of economic indicators from survey samples based on Pareto tail modeling. *Journal of the Royal Statistical Society, Series C*, **62**(2), 271–286.

See Also

`paretoTail`, `reweightOut`, `shrinkOut`

Examples

```
data(eusilc)


## gini coefficient without Pareto tail modeling
gini("eqIncome", weights = "rb050", data = eusilc)


## gini coefficient with Pareto tail modeling

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090,
    groups = eusilc$db030)

# estimate shape parameter
fit <- paretoTail(eusilc$eqIncome, k = ts$k,
    w = eusilc$db090, groups = eusilc$db030)

# replacement of outliers
eqIncome <- replaceOut(fit)
gini(eqIncome, weights = eusilc$rb050)

# replacement of whole tail
eqIncome <- replaceTail(fit)
gini(eqIncome, weights = eusilc$rb050)
```

---

reweightOut *Reweight outliers in the Pareto model*

Description

Reweight observations that are flagged as outliers in a Pareto model for the upper tail of the distribution.

Usage

```
reweightOut(x, ...)

## S3 method for class 'paretoTail'
reweightOut(x, X, w = NULL, ...)
```

Arguments

x	an object of class "paretoTail" (see paretoTail).
...	additional arguments to be passed down.
X	a matrix of binary calibration variables (see calibVars). This is only used if x contains sample weights or if w is supplied.

`w`	a numeric vector of sample weights. This is only used if `x` does not contain sample weights, i.e., if sample weights were not considered in estimating the shape parameter of the Pareto distribution.

Details

If the data contain sample weights, the weights of the outlying observations are set to 1 and the weights of the remaining observations are calibrated according to auxiliary variables. Otherwise, weight 0 is assigned to outliers and weight 1 to other observations.

Value

If the data contain sample weights, a numeric containing the recalibrated weights is returned, otherwise a numeric vector assigning weight 0 to outliers and weight 1 to other observations.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2013) Robust estimation of economic indicators from survey samples based on Pareto tail modeling. *Journal of the Royal Statistical Society, Series C*, **62**(2), 271–286.

See Also

`paretoTail`, `shrinkOut` , `replaceOut`, `replaceTail`

Examples

```
data(eusilc)

## gini coefficient without Pareto tail modeling
gini("eqIncome", weights = "rb050", data = eusilc)

## gini coefficient with Pareto tail modeling
# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090,
    groups = eusilc$db030)
# estimate shape parameter
fit <- paretoTail(eusilc$eqIncome, k = ts$k,
    w = eusilc$db090, groups = eusilc$db030)
# calibration of outliers
w <- reweightOut(fit, calibVars(eusilc$db040))
gini(eusilc$eqIncome, w)
```

rmpg *Relative median at-risk-of-poverty gap*

Description

Estimate the relative median at-risk-of-poverty gap, which is defined as the relative difference between the median equivalized disposable income of persons below the at-risk-of-poverty threshold and the at-risk-of-poverty threshold itself (expressed as a percentage of the at-risk-of-poverty threshold).

Usage

```
rmpg(
  inc,
  weights = NULL,
  sort = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  var = NULL,
  alpha = 0.05,
  na.rm = FALSE,
  ...
)
```

Arguments

`inc`	either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`weights`	optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`sort`	optional; either a numeric vector giving the personal IDs to be used as tie-breakers for sorting, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.
`years`	optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.
`breakdown`	optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value. Note that the same (overall) threshold is used for all domains.

`design` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different strata for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`cluster` optional and only used if `var` is not `NULL`; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`var` a character string specifying the type of variance estimation to be used, or `NULL` to omit variance estimation. See `variance` for possible values.

`alpha` numeric; if `var` is not `NULL`, this gives the significance level to be used for computing the confidence interval (i.e., the confidence level is $1-$`alpha`).

`na.rm` a logical indicating whether missing values should be removed.

`...` if `var` is not `NULL`, additional arguments to be passed to `variance`.

Details

The implementation strictly follows the Eurostat definition.

Value

A list of class `"rmpg"` (which inherits from the class `"indicator"`) with the following components:

`value` a numeric vector containing the overall value(s).

`valueByStratum` a `data.frame` containing the values by domain, or `NULL`.

`varMethod` a character string specifying the type of variance estimation used, or `NULL` if variance estimation was omitted.

`var` a numeric vector containing the variance estimate(s), or `NULL`.

`varByStratum` a `data.frame` containing the variance estimates by domain, or `NULL`.

`ci` a numeric vector or matrix containing the lower and upper endpoints of the confidence interval(s), or `NULL`.

`ciByStratum` a `data.frame` containing the lower and upper endpoints of the confidence intervals by domain, or `NULL`.

`alpha` a numeric value giving the significance level used for computing the confidence interval(s) (i.e., the confidence level is $1-$`alpha`), or `NULL`.

`years` a numeric vector containing the different years of the survey.

`strata` a character vector containing the different domains of the breakdown.

`threshold` a numeric vector containing the at-risk-of-poverty threshold(s).

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat, Luxembourg.

See Also

`arpt`, `variance`

Examples

```
data(eusilc)

# overall value
rmpg("eqIncome", weights = "rb050", data = eusilc)

# values by region
rmpg("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
```

---

ses *Synthetic SES survey data*

Description

This data set is a subset of synthetically generated real Austrian SES (Structural Earnings Survey) data.

Usage

```
data(ses)
```

Format

A data frame with 115691 observations on the following 28 variables.

- `location` geographical location with levels `AT1` (eastern Austria), `AT2` (southern Austria), and `AT3` (western Austria).
- `NACE1` economic branch given in NACE (C - O) 1-digit classification.
- `size` employment size range in 5 categories.
- `economicFinanc` form of economic and financial control (levels `A` = public and financial control, `B` = private control).

payAgreement collective bargaining agreement with levels A = national level pay agreement or interconfederal agreement, B = industry agreement, C = agreement of individual industries in individual regions, D = enterprise or single employer agreement, E = agreement applying only to workers in the local unit, F = any other type of agreement, N = no collective agreement exists

IDunit ID for place of employment.

sex gender with levels female and male.

age age in age classes.

education highest education.

occupation occupation with levels 11 = Legislators and seniors officials, 12 = Corporate managers, 13 = Managers of small enterprises, 21 = Physical, mathematical and engineering science professionals, 22 = Life science and health professionals, 23 = Teaching professionals, 24 = Other professionals, 31 = Physical and engineering science associate professionals, 32 = Life science and health associate professionals, 33 = Teaching associate professionals, 34 = Other associate professionals, 41 = Office clerks, 42 = Customer services clerks, 51 = Personal and protective services workers, 52 = Models, salespersons and demonstrators, 61 = Skilled agricultural and fishery workers, 71 = Extraction and building trades workers, 72 = Metal, machinery and related trades workers, 73 = Precision, handicraft, craft printing and related trades workers, 74 = Other craft and related trades workers, 81 = Stationary plant and related operators, 82 = Machine operators and assemblers, 83 = Drivers and mobile plant operators, 91 = Sales and services elementary occupations, 92 = Agricultural, fishery and related labourers, 93 = Labourers in mining, construction, manufacturing and transport

contract type of contract. Levels A = indefinite duration, employment contract, B = temporary fixed duration C = apprentice.

fullPart full-time working time (FT) or part-time employee (PT).

lengthService The total length of service in the enterprises in the reference month is be based on the number of completed years of service.

weeks the number of weeks in the reference year to which the gross annual earnings relate is mentioned. That is the employee's working time actually paid during the year and should correspond to the actual gross annual earnings.

hoursPaid the number of hours paid in the reference month which means these hours actually paid including all normal and overtime hours worked and remunerated by the employee during the month.

overtimeHours the number of overtime hours paid in the reference month. Overtime hours are those worked in addition to those of the normal working month.

shareNormalHours the share of a full timer's normal hours. The hours contractually worked of a part-time employee are expressed as percentages of the number of normal hours worked by a full-time employee in the local unit.

holiday the annual days of holiday leave (in full days).

notPaid examples of annual bonuses and allowances are Christmas and holiday bonuses, 13th and 14th month payments and productivity bonuses, hence any periodic, irregular and exceptional bonuses and other payments that do not feature every pay period. Besides the main difference between annual earnings and monthly earnings is the inclusion of payments that do not regularly occur in each pay period.

`earningsOvertime` earnings related to overtime.

`paymentsShiftWork` These special payments for shift work are premium payments during the reference month for shirt work, night work or weekend work where they are not treated as overtime.

`earningsMonth` the gross earnings in the reference month covers remuneration in cash paid during the reference month before any tax deductions and social security deductions and social security contributions payable by wage earners and retained by the employer.

`earnings` gross annual earnings in the reference year.

`earningsHour` hourly earnings, being the quotient of monthly earnings and the number of hours paid in the reference month.

`weightsEmployers` sampling weights in the first stage at employer level.

`weightsEmployees` sampling weights corresponding to the second stage at employee level.

`weights` the final sampling weights, which is the product of `weightsEmployers` and `weighsEmployees`.

Details

The Structural Earnings Survey (SES) is conducted in almost all European Countries, and the most important figures are reported to Eurostat. SES is a complex survey of enterprises and establishments with more than 10 employees, NACE C-O, including a large sample of employees. In many countries, a two-stage design is used where in the first stage a stratified sample of enterprises and establishments on NACE 1-digit level, NUTS 1 and employment size range is used, and large enterprises have higher inclusion probabilities. In stage 2, systematic sampling is applied in each enterprise using unequal inclusion probabilities regarding employment size range categories.

The data set in the package consists of enterprise and employees data from 500 places of work. Note that this is a subset of synthetic data set that is simulated from the original Austrian SES data.

Author(s)

Matthias Templ, Karoline Geissler

Source

This is a synthetic data set based on Austrian SES data from 2006.

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

T. Geissberger (2009) Verdienststrukturerhebung 2006, Struktur und Verteilung der Verdienste in Oesterreich, Statistik Austria, ISBN 978-3-902587-97-8.

M. Templ (2012) Comparison of perturbation methods based on pre-defined quality indicators, *UNECE Work Session on Statistical Data Editing*, Tarragona, Spain.

Examples

```
data(ses)
summary(ses)
```

shrinkOut *Shrink outliers in the Pareto model*

Description

Shrink observations that are flagged as outliers in a Pareto model for the upper tail of the distribution to the theoretical quantile used for outlier detection.

Usage

```
shrinkOut(x, ...)

## S3 method for class 'paretoTail'
shrinkOut(x, ...)
```

Arguments

`x`	an object of class `"paretoTail"` (see `paretoTail`).
`...`	additional arguments to be passed down (currently ignored as there are no additional arguments in the only method implemented).

Value

A numeric vector consisting mostly of the original values, but with outlying observations in the upper tail shrunken to the corresponding theoretical quantile of the fitted Pareto distribution.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

See Also

`paretoTail`, `reweightOut`, `replaceOut`, `replaceTail`

Examples

```
data(eusilc)

## gini coefficient without Pareto tail modeling
gini("eqIncome", weights = "rb050", data = eusilc)

## gini coefficient with Pareto tail modeling
# estimate threshold
```

```
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090,
    groups = eusilc$db030)
# estimate shape parameter
fit <- paretoTail(eusilc$eqIncome, k = ts$k,
    w = eusilc$db090, groups = eusilc$db030)
# shrink outliers
eqIncome <- shrinkOut(fit)
gini(eqIncome, weights = eusilc$rb050)
```

`thetaHill` *Hill estimator*

Description

The Hill estimator uses the maximum likelihood principle to estimate the shape parameter of a Pareto distribution.

Usage

```
thetaHill(x, k = NULL, x0 = NULL, w = NULL)
```

Arguments

`x` a numeric vector.

`k` the number of observations in the upper tail to which the Pareto distribution is fitted.

`x0` the threshold (scale parameter) above which the Pareto distribution is fitted.

`w` an optional numeric vector giving sample weights.

Details

The arguments `k` and `x0` of course correspond with each other. If `k` is supplied, the threshold `x0` is estimated with the $n - k$ largest value in `x`, where n is the number of observations. On the other hand, if the threshold `x0` is supplied, `k` is given by the number of observations in `x` larger than `x0`. Therefore, either `k` or `x0` needs to be supplied. If both are supplied, only `k` is used (mainly for back compatibility).

Value

The estimated shape parameter.

Note

The arguments `x0` for the threshold (scale parameter) of the Pareto distribution and `w` for sample weights were introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

Hill, B.M. (1975) A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, **3**(5), 1163–1174.

See Also

`paretoTail`, `fitPareto`, `thetaPDC`, `thetaWML`, `thetaISE`, `minAMSE`

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaHill(eusilc$eqIncome, k = ts$k, w = eusilc$db090)

# using threshold
thetaHill(eusilc$eqIncome, x0 = ts$x0, w = eusilc$db090)
```

`thetaISE` *Integrated squared error (ISE) estimator*

Description

The integrated squared error (ISE) estimator estimates the shape parameter of a Pareto distribution based on the relative excesses of observations above a certain threshold.

Usage

```
thetaISE(x, k = NULL, x0 = NULL, w = NULL, ...)
```

Arguments

`x`	a numeric vector.
`k`	the number of observations in the upper tail to which the Pareto distribution is fitted.
`x0`	the threshold (scale parameter) above which the Pareto distribution is fitted.
`w`	an optional numeric vector giving sample weights.
`...`	additional arguments to be passed to `optimize` (see “Details”).

Details

The arguments k and x0 of course correspond with each other. If k is supplied, the threshold x0 is estimated with the $n - k$ largest value in x, where n is the number of observations. On the other hand, if the threshold x0 is supplied, k is given by the number of observations in x larger than x0. Therefore, either k or x0 needs to be supplied. If both are supplied, only k is used (mainly for back compatibility).

The ISE estimator minimizes the integrated squared error (ISE) criterion with a complete density model. The minimization is carried out using nlm. By default, the starting value is obtained the Hill estimator (see thetaHill). optimize.

Value

The estimated shape parameter.

Note

The arguments x0 for the threshold (scale parameter) of the Pareto distribution and w for sample weights were introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2013) Robust estimation of economic indicators from survey samples based on Pareto tail modeling. *Journal of the Royal Statistical Society, Series C*, **62**(2), 271–286.

Vandewalle, B., Beirlant, J., Christmann, A., and Hubert, M. (2007) A robust estimator for the tail index of Pareto-type distributions. *Computational Statistics & Data Analysis*, **51**(12), 6252–6268.

See Also

paretoTail, fitPareto, thetaPDC, thetaHill

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaISE(eusilc$eqIncome, k = ts$k, w = eusilc$db090)
```

```
# using threshold
thetaISE(eusilc$eqIncome, x0 = ts$x0, w = eusilc$db090)
```

thetaLS *Least squares (LS) estimator*

Description

Estimate the shape parameter of a Pareto distribution using a least squares (LS) approach.

Usage

```
thetaLS(x, k = NULL, x0 = NULL)
```

Arguments

x	a numeric vector.
k	the number of observations in the upper tail to which the Pareto distribution is fitted.
x0	the threshold (scale parameter) above which the Pareto distribution is fitted.

Details

The arguments k and x0 of course correspond with each other. If k is supplied, the threshold x0 is estimated with the $n - k$ largest value in x, where n is the number of observations. On the other hand, if the threshold x0 is supplied, k is given by the number of observations in x larger than x0. Therefore, either k or x0 needs to be supplied. If both are supplied, only k is used (mainly for back compatibility).

Value

The estimated shape parameter.

Note

The argument x0 for the threshold (scale parameter) of the Pareto distribution was introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

Brazauskas, V. and Serfling, R. (2000) Robust estimation of tail parameters for two-parameter Pareto and exponential models via generalized quantile statistics. *Extremes*, **3**(3), 231–249.

Brazauskas, V. and Serfling, R. (2000) Robust and efficient estimation of the tail index of a single-parameter Pareto distribution. *North American Actuarial Journal*, **4**(4), 12–27.

See Also

paretoTail, fitPareto

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaLS(eusilc$eqIncome, k = ts$k)

# using threshold
thetaLS(eusilc$eqIncome, x0 = ts$x0)
```

thetaMoment *Moment estimator*

Description

Estimate the shape parameter of a Pareto distribution based on moments.

Usage

```
thetaMoment(x, k = NULL, x0 = NULL)
```

Arguments

x	a numeric vector.
k	the number of observations in the upper tail to which the Pareto distribution is fitted.
x0	the threshold (scale parameter) above which the Pareto distribution is fitted.

Details

The arguments k and x0 of course correspond with each other. If k is supplied, the threshold x0 is estimated with the $n - k$ largest value in x, where n is the number of observations. On the other hand, if the threshold x0 is supplied, k is given by the number of observations in x larger than x0. Therefore, either k or x0 needs to be supplied. If both are supplied, only k is used (mainly for back compatibility).

Value

The estimated shape parameter.

Note

The argument x0 for the threshold (scale parameter) of the Pareto distribution was introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

Dekkers, A.L.M., Einmahl, J.H.J. and de Haan, L. (1989) A moment estimator for the index of an extreme-value distribution. *The Annals of Statistics*, **17**(4), 1833–1855.

See Also

paretoTail, fitPareto

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaMoment(eusilc$eqIncome, k = ts$k)

# using threshold
thetaMoment(eusilc$eqIncome, x0 = ts$x0)
```

thetaPDC *Partial density component (PDC) estimator*

Description

The partial density component (PDC) estimator estimates the shape parameter of a Pareto distribution based on the relative excesses of observations above a certain threshold.

Usage

```
thetaPDC(x, k = NULL, x0 = NULL, w = NULL, ...)
```

Arguments

x	a numeric vector.
k	the number of observations in the upper tail to which the Pareto distribution is fitted.
x0	the threshold (scale parameter) above which the Pareto distribution is fitted.
w	an optional numeric vector giving sample weights.
...	additional arguments to be passed to optimize (see “Details”).

Details

The arguments k and x0 of course correspond with each other. If k is supplied, the threshold x0 is estimated with the $n - k$ largest value in x, where n is the number of observations. On the other hand, if the threshold x0 is supplied, k is given by the number of observations in x larger than x0. Therefore, either k or x0 needs to be supplied. If both are supplied, only k is used (mainly for back compatibility).

The PDC estimator minimizes the integrated squared error (ISE) criterion with an incomplete density mixture model. The minimization is carried out using nlm. By default, the starting value is obtained with the Hill estimator (see thetaHill). optimize.

Value

The estimated shape parameter.

Note

The arguments x0 for the threshold (scale parameter) of the Pareto distribution and w for sample weights were introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

A. Alfons, M. Templ, P. Filzmoser (2013) Robust estimation of economic indicators from survey samples based on Pareto tail modeling. *Journal of the Royal Statistical Society, Series C*, **62**(2), 271–286.

Vandewalle, B., Beirlant, J., Christmann, A., and Hubert, M. (2007) A robust estimator for the tail index of Pareto-type distributions. *Computational Statistics & Data Analysis*, **51**(12), 6252–6268.

See Also

`paretoTail`, `fitPareto`, `thetaISE`, `thetaHill`

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaPDC(eusilc$eqIncome, k = ts$k, w = eusilc$db090)

# using threshold
thetaPDC(eusilc$eqIncome, x0 = ts$x0, w = eusilc$db090)
```

`thetaQQ` *QQ-estimator*

Description

Estimate the shape parameter of a Pareto distribution using a quantile-quantile approach.

Usage

```
thetaQQ(x, k = NULL, x0 = NULL)
```

Arguments

`x`	a numeric vector.
`k`	the number of observations in the upper tail to which the Pareto distribution is fitted.
`x0`	the threshold (scale parameter) above which the Pareto distribution is fitted.

Details

The arguments k and x0 of course correspond with each other. If k is supplied, the threshold x0 is estimated with the $n - k$ largest value in x, where n is the number of observations. On the other hand, if the threshold x0 is supplied, k is given by the number of observations in x larger than x0. Therefore, either k or x0 needs to be supplied. If both are supplied, only k is used (mainly for back compatibility).

Value

The estimated shape parameter.

Note

The argument x0 for the threshold (scale parameter) of the Pareto distribution was introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

Kratz, M.F. and Resnick, S.I. (1996) The QQ-estimator and heavy tails. *Stochastic Models*, **12**(4), 699–724.

See Also

paretoTail, fitPareto

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaQQ(eusilc$eqIncome, k = ts$k)

# using threshold
thetaQQ(eusilc$eqIncome, x0 = ts$x0)
```

`thetaTM` *Trimmed mean estimator*

Description

Estimate the shape parameter of a Pareto distribution using a trimmed mean approach.

Usage

```
thetaTM(x, k = NULL, x0 = NULL, beta = 0.05)
```

Arguments

`x` a numeric vector.

`k` the number of observations in the upper tail to which the Pareto distribution is fitted.

`x0` the threshold (scale parameter) above which the Pareto distribution is fitted.

`beta` A numeric vector of length two giving the trimming proportions for the lower and upper end of the tail, respectively. If a single numeric value is supplied, it is recycled.

Details

The arguments `k` and `x0` of course correspond with each other. If `k` is supplied, the threshold `x0` is estimated with the $n - k$ largest value in `x`, where n is the number of observations. On the other hand, if the threshold `x0` is supplied, `k` is given by the number of observations in `x` larger than `x0`. Therefore, either `k` or `x0` needs to be supplied. If both are supplied, only `k` is used (mainly for back compatibility).

Value

The estimated shape parameter.

Note

The argument `x0` for the threshold (scale parameter) of the Pareto distribution was introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

Brazauskas, V. and Serfling, R. (2000) Robust estimation of tail parameters for two-parameter Pareto and exponential models via generalized quantile statistics. *Extremes*, **3**(3), 231–249.

Brazauskas, V. and Serfling, R. (2000) Robust and efficient estimation of the tail index of a single-parameter Pareto distribution. *North American Actuarial Journal*, **4**(4), 12–27.

See Also

paretoTail, fitPareto

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaTM(eusilc$eqIncome, k = ts$k)

# using threshold
thetaTM(eusilc$eqIncome, x0 = ts$x0)
```

---

thetaWML *Weighted maximum likelihood estimator*

Description

Estimate the shape parameter of a Pareto distribution using a weighted maximum likelihood approach.

Usage

```
thetaWML(
  x,
  k = NULL,
  x0 = NULL,
  weight = c("residuals", "probability"),
  const,
  bias = TRUE,
  ...
)
```

Arguments

x	a numeric vector.
k	the number of observations in the upper tail to which the Pareto distribution is fitted.
x0	the threshold (scale parameter) above which the Pareto distribution is fitted.

`weight` a character string specifying the weight function to be used. If `"residuals"` (the default), the weight function is based on standardized residuals. If `"probability"`, probability based weighting is used. Partial string matching allows these names to be abbreviated.

`const` Tuning constant(s) that control the robustness of the method. If `weight="residuals"`, a single numeric value is required (the default is 2.5). If `weight="probability"`, a numeric vector of length two must be supplied (a single numeric value is recycled; the default is 0.005 for both tuning parameters). See the references for more details.

`bias` a logical indicating whether bias correction should be applied.

`...` additional arguments to be passed to `uniroot` (see "Details").

Details

The arguments `k` and `x0` of course correspond with each other. If `k` is supplied, the threshold `x0` is estimated with the $n - k$ largest value in `x`, where n is the number of observations. On the other hand, if the threshold `x0` is supplied, `k` is given by the number of observations in `x` larger than `x0`. Therefore, either `k` or `x0` needs to be supplied. If both are supplied, only `k` is used (mainly for back compatibility).

The weighted maximum likelihood estimator belongs to the class of M-estimators. In order to obtain the estimate, the root of a certain function needs to be found, which is implemented using `uniroot`.

Value

The estimated shape parameter.

Note

The argument `x0` for the threshold (scale parameter) of the Pareto distribution was introduced in version 0.2.

Author(s)

Andreas Alfons and Josef Holzer

References

Dupuis, D.J. and Morgenthaler, S. (2002) Robust weighted likelihood estimators with an application to bivariate extreme value problems. *The Canadian Journal of Statistics*, **30**(1), 17–36.

Dupuis, D.J. and Victoria-Feser, M.-P. (2006) A robust prediction error criterion for Pareto modelling of upper tails. *The Canadian Journal of Statistics*, **34**(4), 639–658.

See Also

`paretoTail`, `fitPareto`

Examples

```
data(eusilc)
# equivalized disposable income is equal for each household
# member, therefore only one household member is taken
eusilc <- eusilc[!duplicated(eusilc$db030),]

# estimate threshold
ts <- paretoScale(eusilc$eqIncome, w = eusilc$db090)

# using number of observations in tail
thetaWML(eusilc$eqIncome, k = ts$k)

# using threshold
thetaWML(eusilc$eqIncome, x0 = ts$x0)
```

---

utils *Utility functions for indicators on social exclusion and poverty*

Description

Test for class, print and take subsets of indicators on social exclusion and poverty.

Usage

```
is.indicator(x)

is.arpr(x)

is.qsr(x)

is.rmpg(x)

is.gini(x)

is.prop(x)

is.gpg(x)

## S3 method for class 'indicator'
print(x, ...)

## S3 method for class 'arpr'
print(x, ...)

## S3 method for class 'rmpg'
print(x, ...)
```

```
## S3 method for class 'indicator'
subset(x, years = NULL, strata = NULL, ...)

## S3 method for class 'arpr'
subset(x, years = NULL, strata = NULL, ...)

## S3 method for class 'rmpg'
subset(x, years = NULL, strata = NULL, ...)
```

Arguments

`x` for `is.xyz`, any object to be tested. The `print` and `subset` methods are called by the generic functions if an object of the respective class is supplied.

`...` additional arguments to be passed to and from methods.

`years` an optional numeric vector giving the years to be extracted.

`strata` an optional vector giving the domains of the breakdown to be extracted.

Value

`is.indicator` returns `TRUE` if `x` inherits from class `"indicator"` and `FALSE` otherwise.

`is.arpr` returns `TRUE` if `x` inherits from class `"arpr"` and `FALSE` otherwise.

`is.qsr` returns `TRUE` if `x` inherits from class `"qsr"` and `FALSE` otherwise.

`is.rmpg` returns `TRUE` if `x` inherits from class `"rmpg"` and `FALSE` otherwise.

`is.gini` returns `TRUE` if `x` inherits from class `"gini"` and `FALSE` otherwise.

`is.gini` returns `TRUE` if `x` inherits from class `"gini"` and `FALSE` otherwise.

`print.indicator`, `print.arpr` and `print.rmpg` return `x` invisibly.

`subset.indicator`, `subset.arpr` and `subset.rmpg` return a subset of `x` of the same class.

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

See Also

`arpr`, `qsr`, `rmpg`, `gini`, `gpg`

Examples

```
data(eusilc)

# at-risk-of-poverty rate
a <- arpr("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
print(a)
is.arpr(a)
is.indicator(a)
```

```
subset(a, strata = c("Lower Austria", "Vienna"))

# quintile share ratio
q <- qsr("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
print(q)
is.qsr(q)
is.indicator(q)
subset(q, strata = c("Lower Austria", "Vienna"))

# relative median at-risk-of-poverty gap
r <- rmpg("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
print(r)
is.rmpg(r)
is.indicator(r)
subset(r, strata = c("Lower Austria", "Vienna"))

# Gini coefficient
g <- gini("eqIncome", weights = "rb050",
    breakdown = "db040", data = eusilc)
print(g)
is.gini(g)
is.indicator(g)
subset(g, strata = c("Lower Austria", "Vienna"))
```

variance — *Variance and confidence intervals of indicators on social exclusion and poverty*

Description

Compute variance and confidence interval estimates of indicators on social exclusion and poverty.

Usage

```
variance(
  inc,
  weights = NULL,
  years = NULL,
  breakdown = NULL,
  design = NULL,
  cluster = NULL,
  data = NULL,
  indicator,
  alpha = 0.05,
  na.rm = FALSE,
  type = "bootstrap",
  gender = NULL,
  method = NULL,
  ...
)
```

Arguments

`inc` either a numeric vector giving the equivalized disposable income, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`weights` optional; either a numeric vector giving the personal sample weights, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`years` optional; either a numeric vector giving the different years of the survey, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, values are computed for each year.

`breakdown` optional; either a numeric vector giving different domains, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`. If supplied, the values for each domain are computed in addition to the overall value.

`design` optional; either an integer vector or factor giving different strata for stratified sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`cluster` optional; either an integer vector or factor giving different clusters for cluster sampling designs, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`data` an optional `data.frame`.

`indicator` an object inheriting from the class `"indicator"` that contains the point estimates of the indicator (see `arpr`, `qsr`, `rmpg` or `gini`).

`alpha` a numeric value giving the significance level to be used for computing the confidence interval(s) (i.e., the confidence level is $1-$`alpha`), or `NULL`.

`na.rm` a logical indicating whether missing values should be removed.

`type` a character string specifying the type of variance estimation to be used. Currently, only `"bootstrap"` is implemented for variance estimation based on bootstrap resampling (see `bootVar`).

`gender` either a numeric vector giving the gender, or (if `data` is not `NULL`) a character string, an integer or a logical vector specifying the corresponding column of `data`.

`method` a character string specifying the method to be used (only for `gpg`). Possible values are `"mean"` for the mean, and `"median"` for the median. If weights are provided, the weighted mean or weighted median is estimated.

`...` additional arguments to be passed to `bootVar`.

Details

This is a wrapper function for computing variance and confidence interval estimates of indicators on social exclusion and poverty.

Value

An object of the same class as `indicator` is returned. See `arpr`, `qsr`, `rmpg` or `gini` for details on the components.

Author(s)

Andreas Alfons

References

A. Alfons and M. Templ (2013) Estimation of Social Exclusion Indicators from Complex Surveys: The R Package **laeken**. *Journal of Statistical Software*, **54**(15), 1–25. doi:10.18637/jss.v054.i15

See Also

`bootVar`, `arpr`, `qsr`, `rmpg`, `gini`

Examples

```
data(eusilc)
a <- arpr("eqIncome", weights = "rb050", data = eusilc)

## naive bootstrap
variance("eqIncome", weights = "rb050", design = "db040",
    data = eusilc, indicator = a, R = 50,
    bootType = "naive", seed = 123)

## bootstrap with calibration
variance("eqIncome", weights = "rb050", design = "db040",
    data = eusilc, indicator = a, R = 50,
    X = calibVars(eusilc$db040), seed = 123)
```

`weightedMean` *Weighted mean*

Description

Compute the weighted mean.

Usage

```
weightedMean(x, weights = NULL, na.rm = FALSE)
```

Arguments

`x`	a numeric vector.
`weights`	an optional numeric vector giving the sample weights.
`na.rm`	a logical indicating whether missing values in `x` should be omitted.

Details

This is a simple wrapper function calling `weighted.mean` if sample weights are supplied and `mean` otherwise.

Value

The weighted mean of values in `x` is returned.

Author(s)

Andreas Alfons

See Also

`incMean`

Examples

```
data(eusilc)
weightedMean(eusilc$eqIncome, eusilc$rb050)
```

`weightedMedian` *Weighted median*

Description

Compute the weighted median (Eurostat definition).

Usage

```
weightedMedian(x, weights = NULL, sorted = FALSE, na.rm = FALSE)
```

Arguments

`x`	a numeric vector.
`weights`	an optional numeric vector giving the sample weights.
`sorted`	a logical indicating whether the observations in `x` are already sorted.
`na.rm`	a logical indicating whether missing values in `x` should be omitted.

Details

The implementation strictly follows the Eurostat definition.

Value

The weighted median of values in `x` is returned.

Author(s)

Andreas Alfons and Matthias Templ

References

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat.

See Also

arpt, incMedian, weightedQuantile

Examples

```
data(eusilc)
weightedMedian(eusilc$eqIncome, eusilc$rb050)
```

---

weightedQuantile *Weighted quantiles*

Description

Compute weighted quantiles (Eurostat definition).

Usage

```
weightedQuantile(
  x,
  weights = NULL,
  probs = seq(0, 1, 0.25),
  sorted = FALSE,
  na.rm = FALSE
)
```

Arguments

x	a numeric vector.
weights	an optional numeric vector giving the sample weights.
probs	numeric vector of probabilities with values in $[0, 1]$.
sorted	a logical indicating whether the observations in x are already sorted.
na.rm	a logical indicating whether missing values in x should be omitted.

Details

The implementation strictly follows the Eurostat definition.

Value

A numeric vector containing the weighted quantiles of values in `x` at probabilities `probs` is returned. Unlike `quantile`, this returns an unnamed vector.

Author(s)

Andreas Alfons and Matthias Templ

References

Working group on Statistics on Income and Living Conditions (2004) Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. *EU-SILC 131-rev/04*, Eurostat.

See Also

`incQuintile`, `weightedMedian`

Examples

```
data(eusilc)
weightedQuantile(eusilc$eqIncome, eusilc$rb050)
```

Index