Package ‘dineq’

October 13, 2022

Type Package
Title Decomposition of (Income) Inequality
Version 0.1.0
Date 2018-06-19
Author René Schulenberg
Maintainer René Schulenberg <reneschulenberg@gmail.com>
Description Decomposition of (income) inequality by population sub groups.
   For a decomposition on a single variable the mean log deviation can be used
   For a decomposition on multiple variables a regression based technique can be
   used (see Fields (2003) <DOI:10.1016/s0147-9121(03)22001-x>).
   Recentered influence function regression for marginal effects of the (income
   or wealth) distribution (see Firpo et al. (2009) <DOI:10.3982/ECTA6822>).
   Some extensions to inequality functions to handle weights and/or missings.
Depends R (>= 2.10)
Imports boot (>= 1.3-20), Hmisc (>= 4.0-3)
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2018-06-27 15:30:30 UTC

R topics documented:

dineq_change_rb .......................................................... 2
dineq_rb ................................................................. 4
gini.wtd .............................................................. 6
gini_decomp ............................................................ 7
mex_inc_2008 ............................................................. 8
dineq_change_rb

Decomposition of the change in inequality into multiple characteristics, divided by a price and a quantity effect.

Usage

dineq_change_rb(formula1, weights1 = NULL, data1, formula2, weights2 = NULL, data2)

Arguments

formula1: an object of class "formula" (or one that can be coerced to that class) for the first year/dataset: a symbolic description of the model to be fitted in the ordinary least squares regression.

weights1: an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. Should be inside selected data frame in the function and between quotation marks.

data1: a data frame containing the variables for the first year/dataset in the model.

formula2: an object of class "formula" (or one that can be coerced to that class) for the first year/dataset: a symbolic description of the model to be fitted in the ordinary least squares regression.

weights2: an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. Should be inside selected data frame in the function and between quotation marks.

data2: a data frame containing the variables for the first year/dataset in the model.
Details

This function uses a multivariate regression-based decomposition method. Multiple characteristics can be added to the function in order to calculate the contribution of each individual variable (including the residual) to the change of the inequality. For instance socio-economic, demographic and geographic characteristics (such as age, household composition, gender, region, education) of the household or the individual can be added.

The change decomposition is divided into a price and a quantity effect for each characteristic. The quantity effect is caused by changes in the relative size of subgroups (for instance: a higher percentage of elderly households). The price effect is caused by a change in the influence of the characteristic on the dependent variable (for instance a higher income for the elderly households).

It uses a logarithmic transformation of the values of the dependent variable. Therefore it cannot handle negative or zero values. Those are excluded from the computation in this function.

The decomposition can only be used on the variance of log income.

The main difference with the decomposition of the change of the mean log deviation is that multiple characteristics can be analyzed at the same time. While the decomposition function only analyze one characteristic at the same time.

The function uses two datasets for both years to compare. Pay attention that characteristics should be the same (although can be named differently) and in the same order in the formula.

Value

a list with the results of the decomposition and the parts used for the decomposition, containing the following components:

- **attention** optional note on the difference in the input.
- **variance_logincome** the values of the variance of log income of both years/datasets and difference between both.
- **decomposition_inequality** the (relative) decomposition of the inequality of both years/datasets into the different variables. See function 'rb_decomp'.
- **decomposition_change_absolute** decomposition of the change in the variance of log income into the different variables and residual split into price and quantity effects. Adds up to the absolute change in variance of log income.
- **decomposition_change_relative** decomposition of the change in the variance of log income into the different variables and residual split into price and quantity effects. Adds up to 100 percent.
- **notes** number of zero or negative observations in both data sets/years. The function uses a logarithmic transformation of x as input for the regression. Therefore these observations are deleted from the analysis

References


**See Also**

dineq_rb

**Examples**

# Decomposition of the change in income inequality into 4 variables using the Mexican Income data set
data(mex_inc_2008)

inequality_change <- dineq_change_rb(formula1=income~hh_structure+education+domicile_size+age_cat, weights1="factor",data1=mex_inc_2008, formula2=income~hh_structure+education+domicile_size+age_cat, weights2="factor",data2=mex_inc_2016)

# Selection of output: change in variance of log income decomposed in variables split into price and quantity effect and residual.
inequality_change["decomposition_change_absolute"]

# Selection of output: relative change in variance of log income decomposed in variables split into price and quantity effect and residual. Because of negative change in variance of log income, the negative contribution of education (quantity) becomes a positive number.
inequality_change["decomposition_change_relative"]

---

dineq_rb  
*Regression-based decomposition of inequality*

**Description**

Decomposition of (income) inequality into multiple characteristics. A regression-based decomposition method is used.

**Usage**

dineq_rb(formula, weights = NULL, data)

**Arguments**

- **formula**: an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted in the ordinary least squares regression.
- **weights**: an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector. Should be inside selected data frame in the function and between quotation marks.
- **data**: a data frame containing the variables in the model.
Details

This function uses a multivariate regression-based decomposition method. Multiple variables can be added to the function in order to calculate the contribution of each individual variable (including the residual) to the inequality. For instance socio-economic, demographic and geographic characteristics (such as age, household composition, gender, region, education) of the household or the individual can be added.

This decomposition can be used on a broad range of inequality measure, like Gini, Theil, mean log deviation, Atkinson index and variance of log income.

It uses a logarithmic transformation of the values of the dependent variable. Therefore it cannot handle negative or zero values. Those are excluded from the computation in this function.

The main difference with the decomposition of the mean log deviation or Gini coefficient is that multiple characteristics can be analyzed at the same time. While the other decomposition functions only analyze one characteristic at the same time.

Value

a list with the results of the decomposition, containing the following components:

- inequality_measures
  the values of 4 inequality measures: gini, mean log deviation, theil and variance of log income
- decomposition_inequality
  the (relative) decomposition of the inequality into the different variables
- regression_results
  results of the ols regression which is used to make the decomposition of inequality
- note
  number of zero or negative observations. The function uses a logarithmic transformation of x as input for the regression. Therefore these observations are deleted from the analysis

References


See Also

dineq_change_rb

Examples

#Decomposition of the income inequality into 4 variables using Mexican Income data set:
data(mex_inc_2008)
inequality_decomp <- dineq_rb(income~hh_structure+education+domicile_size+age_cat, weights="factor", data=mex_inc_2008)

#selection of the output: decomposition of the inequality into the contribution of the different variables and residual (adds up to 100 percent)
inequality_decomp["decomposition_inequality"]

---

<table>
<thead>
<tr>
<th>gini.wtd</th>
<th>Gini coefficient</th>
</tr>
</thead>
</table>

**Description**

Returns the (optional weighted) Gini coefficient for a vector.

**Usage**

```r
gini.wtd(x, weights = NULL)
```

**Arguments**

- `x` a numeric vector containing at least non-negative elements.
- `weights` an optional vector of weights of `x` to be used in the computation of the Gini coefficient. Should be NULL or a numeric vector.

**Details**

The Gini coefficient is a measure of inequality among values of a distribution. The most used single measure for income inequality. The coefficient can theoretically range between 0 and 1, with 1 being the highest possible inequality (for instance: 1 person in a society has all income; the others none). But coefficients that are negative or greater than 1 are also possible because of negative values in the distribution. Compared to other measures of inequality, the Gini coefficient is especially sensitive for changes in the middle of the distribution.

Extension of the gini function in reldist package in order to handle missings.

**Value**

The value of the Gini coefficient.

**Source**


**References**


Examples

# calculate Gini coefficient using Mexican Income data set
data(mex_inc_2008)

# unweighted Gini coefficient:
gini.wtd(mex_inc_2008$income)

# weighted Gini coefficient:
gini.wtd(x=mex_inc_2008$income, weights=mex_inc_2008$factor)

\[
gini_{\text{decomp}}
\]
\[
\text{Decomposition of the Gini coefficient}
\]

Description

Decomposes the Gini coefficient into population subgroups. Distinction is made by between and within group inequality and an overlap (interaction) term.

Usage

gini_decomp(x, z, weights = NULL)

Arguments

- **x**  
a numeric vector containing at least non-negative elements.
- **z**  
a factor containing the population sub groups.
- **weights**  
an optional vector of weights of x to be used in the computation of the decomposition. Should be NULL or a numeric vector.

Details

The decomposition of the Gini coefficient by between and within group inequality. In most cases there is an overlap of the distribution of both groups. Consequence is that between and within group inequality doesn’t add up to the total Gini coefficient. In those cases there is an overlap term. Also referred to as interaction effect.

Within group inequality is calculated by using the Gini coefficient for each sub group. Between group inequality by using the gini coefficient of the average of both sub groups.

Value

a list with the results of the decomposition and the parts used for the decomposition, containing the following components:

- **gini_decomp**  
a list containing the decomposition: gini_total (value of the gini coefficient of x),
gini_within (value of within-group inequality), gini_between (value of between-group inequality) and gini_overlap (value of overlap in inequality)
### Description

Selection of Mexican income (survey) data and household characteristic for 2008. Extracted from ENIGH (Household Income and Expenditure Survey).

### References


### See Also

- `mld_decomp`

### Examples

```r
# Decomposition of the gini coefficient by level of education using Mexican Income data set
data(mex_inc_2008)
education_decomp <- gini_decomp(x=mex_inc_2008$income, z=mex_inc_2008$education, weights=mex_inc_2008$factor)

# Complete output
education_decomp

# Selected output: decomposition into between- and within-group inequality and overlap (interaction)
education_decomp["gini_decomp"]
```
mex_inc_2008

Usage

data(mex_inc_2008)

Format

A data frame containing 5000 observations and 8 variables (a selection from the original).

- **hh_number**  Household ID.
- **factor**  Population inflating weights.
- **income**  Household income.
- **hh_structure**  Household structure, factor with levels unipersonal, nuclear, ampliado, compuesto and coresidente.
- **education**  Highest achieved education of the head of the household, factor with levels Sin instruccion, Preescolar, Primaria incompleta, Primaria completa, Secundaria incompleta, Secundaria completa, Preparatoria incompleta, Preparatoria completa, Profesional incompleta, Profesional completa, Posgrado.
- **domicile_size**  Population of domicile, factor with levels <2500, 2500-15000, 15000-100000, >100000.
- **age**  age (integer) of the head of the household.
- **age_cat**  age (categorical) of the head of the household, factor with levels <25, 25-34, 35-44, 45-54, 55-64, 65-74, >=75.

Details

This data set is a selection of the original dataset of the National Institute of Statistics and Geography in Mexico (INEGI). The original contains 29468 observations and 129 variables with information on the income and household characteristics in Mexico. This selection is only meant to be used as a calculation example for the functions in this package. Results will not represent the correct information on the Mexican situation.

Source


References

**mex_inc_2016**

**Mexican income data 2016**

**Description**

Selection of Mexican income (survey) data and household characteristic for 2016. Extracted from ENIGH (Household Income and Expenditure Survey).

**Usage**

```r
data(mex_inc_2016)
```

**Format**

A data frame containing 5000 observations and 8 variables (a selection from the original).

- **hh_number**  Household ID.
- **factor**  Population inflating weights.
- **income**  Household income.
- **hh_structure**  Household structure, factor with levels unipersonal, nuclear, ampliado, compuesto and coresidente.
- **education**  Highest achieved education of the head of the household, factor with levels Sin instruccion, Preescolar, Primaria incompleta, Primaria completa, Secundaria incompleta, Secundaria completa, Preparatoria incompleta, Preparatoria completa, Profesional incompleta, Profesional completa, Posgrado.
- **domicile_size**  Population of domicile, factor with levels <2500, 2500-15000, 15000-100000, >100000.
- **age**  age (integer) of the head of the household.
- **age_cat**  age (categorical) of the head of the household, factor with levels <25, 25-34, 35-44, 45-54, 55-64, 65-74, >=75.

**Details**

This data set is a selection of the original dataset of the National Institute of Statistics and Geography in Mexico (INEGI). The original contains 70311 observations and 127 variables with information on the income and household characteristics in Mexico. This selection is only meant to be used as a calculation example for the functions in this package. Results will not represent the correct information on the Mexican situation.

**Source**


**References**

**mld.wtd**

**Mean log deviation**

**Description**
Returns the (optional weighted) mean log deviation for a vector.

**Usage**

```
mld.wtd(x, weights = NULL)
```

**Arguments**

- `x` a numeric vector containing at least non-negative elements.
- `weights` an optional vector of weights of `x` to be used in the computation of the mean log deviation. Should be NULL or a numeric vector.

**Details**

The mean log deviation is a measure of inequality among values of a distribution. It is a member of the Generalized Entropy Measures. Also referred to as GE(0). A value of zero is the lowest possible inequality. The measure does not have an upper bound for the highest inequality. It uses a logarithmic transformation of the values of the distribution. Therefore it cannot handle negative or zero values. Those are excluded from the computation in this function. The mean log deviation is more sensitive for changes in the lower tail of the distribution.

Extension of the calcGEI function in IC2 package in order to handle missings.

**Value**

the value of the mean log deviation index.

**Source**


**References**


Examples

# calculate mean log deviation using Mexican Income data set
data(mex_inc_2008)

# unweighted mean log deviation:
mld.wtd(mex_inc_2008$income)

# weighted mean log deviation:
mld.wtd(x=mex_inc_2008$income, weights=mex_inc_2008$factor)

---

mld_change

Decomposition of the change of the mean log deviation

Description

Decomposes the change of the mean log deviation between two years/data sets into population subgroups.

Usage

mld_change(x1, z1, weights1 = NULL, x2, z2, weights2 = NULL)

Arguments

- **x1**: a numeric vector for the first year/dataset containing at least non-negative elements.
- **z1**: a factor for the first year/dataset containing the population subgroups.
- **weights1**: an optional vector of weights of x for the first year/dataset to be used in the computation of the decomposition. Should be NULL or a numeric vector.
- **x2**: a numeric vector for the second year/dataset containing at least non-negative elements.
- **z2**: a factor for the second year/dataset containing the population subgroups.
- **weights2**: an optional vector of weights of x for the second year/dataset to be used in the computation of the decomposition. Should be NULL or a numeric vector.

Details

The change of the mean log deviation can be decomposed into three components: inequality changes between and within groups and changes in the relative sizes of the groups. The change of between group inequality is measures by a change in the relative income of the subgroups. The change of within group inequality by adding up all changes in mean log deviation within the subgroups. And the contribution of changes in relative population size effects the change on both the within and between group components. For the relative contributions those two are added together.

This method is introduced by Mookherjee and Shorrocks. It is an accurate approximation of the exact decomposition. It uses a logarithmic transformation of the values of the distribution. Therefore it cannot handle negative or zero values. Those are excluded from the computation in this function.
**Value**

A list with the results of the decomposition and the parts used for the decomposition, containing the following components:

- `mld_data1`: the value of the mean log deviation index of x for the first year/dataset, and the decomposition into within-group and between-group inequality.
- `mld_data2`: the value of the mean log deviation index of x for the second year/dataset, and the decomposition into within-group and between-group inequality.
- `mld_difference`: the difference between the mean log deviation and the decomposition between the second and first year/dataset.
- `absolute_contributions_difference`: decomposition of the absolute change in inequality into: within group changes, group size changes (split into the effect of within and between group components) and between group changes.
- `relative_contributions_difference`: decomposition of the change in inequality into relative contributions of: within group changes, group size changes and between group changes. Adds up to 100 percent (or -100 percent for negative change).

**Note**

Number of zero or negative observations in both datasets. The mean log deviation uses a logarithmic transformation of x. Therefore these observations are deleted from the analysis.

**References**


**See Also**

- `mld_decomp`

**Examples**

```r
# Decomposition of the change in mean log deviation by level of education using
# Mexican Income data set
data(mex_inc_2008)

change_education <- mld_change(x1=mex_inc_2008$income, z1=mex_inc_2008$education, weights1=mex_inc_2008$factor, x2=mex_inc_2016$income, z2=mex_inc_2016$education, weights2=mex_inc_2016$factor)

# Selection of the output: decomposition of the change into within- and between-group
# contribution and change in size of groups (adds up to 100 percent)
change_education["relative_contributions_difference"]
```

Description

Decomposes the mean log deviation into non overlapping population subgroups. Distinction is made by between and within group inequality.

Usage

mld_decomp(x, z, weights = NULL)

Arguments

x a numeric vector containing at least non-negative elements.
z a factor containing the population subgroups.
weights an optional vector of weights of x to be used in the computation of the decomposition. Should be NULL or a numeric vector.

Details

The decomposition of the mean log deviation by between and within group inequality. Within group inequality is calculated by using the mean log deviation for each subgroup. Between group inequality by the mean log deviation of the average of both sub groups.

It uses a logarithmic transformation of the values of the distribution. Therefore it cannot handle negative or zero values. Those are excluded from the computation in this function.

Based on calcGEI function in IC2 package. Handles missings.

Value

a list with the results of the decomposition and the parts used for the decomposition, containing the following components:

mld_decomp a list containing the decomposition: mld_total (value of the mean log deviation index of x) mld_within (value of within-group inequality) and mld_between (value of between-group inequality)
mld_group a list containing mld_group (the mean log deviations of the different subgroups) and mld_group_contribution(the contribution of the subgroups to the total within-group inequality: adds up to mld_within)mld_decomp a list containing the means of x: mean_total (value of the mean of x of all subgroups combined) and mean_group (value of the mean of x of the individual subgroups) inequality) and mld_between (value of between-group inequality)
share_groups the distribution of the subgroups z
share_income_groups the distribution of vector x by subgroups z
ntiles.wtd

number_cases  a list containing the number of cases in total, by subgroup (weighted and unweighted): n_unweighted (total number of unweighted x), n_weighted (total number of weighted x), n_group_unweighted (number of unweighted x by subgroup z), n_group_weighted (number of weighted x by subgroup z)

note  number of zero or negative observations. The mean log deviation uses a logarithmic transformation of x. Therefore these observations are deleted from the analysis.

Source


References


See Also

mld_change gini_decomp

Examples

#Decomposition of mean log deviation by level of education using Mexican Income data set
data(mex_inc_2008)
education_decomp <- mld_decomp(x=mex_inc_2008$income,z=mex_inc_2008$education, weights=mex_inc_2008$factor)

#complete output
education_decomp

#Selected output: decomposition into between- and within-group inequality
education_decomp["mld_decomp"]

ntiles.wtd  Weighted tiles

Description

Breaks input vector into n groups. Returns the (optional weighted) tile of an individual observation in vector x.
Usage

ntiles.wtd(x, n, weights = NULL)

Arguments

x a numeric vector for which the quantiles are computed. Missing values are left as missing.
n the number of desired sub groups to break vector x into.
weights an optional vector of weights of x to be used in the computation of the tiles. Should be NULL or a numeric vector.

Details

Breaks vector x into n sub groups. The main difference with other tile functions (for instance ntile from dplyr) is that those functions break up vector x in exact equal size sub groups. Observations with the same value can end up in different tiles. In this function, observations with the same value always end up in the same tile, therefore sub groups may have different sizes. Especially when the weights argument is used. For a weighted tile function with the same group size, see for instance weighted_ntile from the grattan package.

When using a short-length vector (compared to the number of tiles) or with high variance weights, output may be different than anticipated.

Value

A vector of integers corresponding to the quantiles of vector x.

Examples

#Break up the income variable in the Mexican Income data set into 10 groups (tiles)
data(mex_inc_2008)

#unweighted tiles:
q <- ntiles.wtd(x=mex_inc_2008$income, n=10)

#weighted tiles:
qw <- ntiles.wtd(x=mex_inc_2008$income, n=10, weights=mex_inc_2008$factor)

polar.wtd

Polarization index
**Description**

Returns the (possibly weighted) polarization index for a vector. The Wolfson index of bipolarization is used.

A bipolarized (income) distribution has fewer observations in the middle and more in lower and/or higher part of the distribution. The regular measures of inequality (like the gini coefficient) does not give information about the polarization of the distribution. This Polarization index computes the level of bipolarization of the distribution. The concept is closely related to the Lorenz curve and therefore the scalar measure is also related to the Gini coefficient. A lower number means a lower level of polarization.

Extension of the polar.aff function in affluence-index package. Option of weighting the index is included.

**Usage**

```r
polar.wtd(x, weights = NULL)
```

**Arguments**

- `x`: a numeric vector.
- `weights`: an optional vector of weights of `x` to be used in the computation of the Polarization index. Should be `NULL` or a numeric vector.

**Value**

The value of the Wolfson polarization index.

**Source**


**References**


**Examples**

```r
#calculate Polarization Index using Mexican Income data set
data(mex_inc_2008)

#unweighted Polarization Index:
polar.wtd(mex_inc_2008$income)

#weighted Polarization Index:
polar.wtd(x=mex_inc_2008$income, weights=mex_inc_2008$factor)
```
Recentered influence function (RIF)

Description

Returns the (optional weighted) recentered influence function of a distributional statistic.

Usage

rif(x, weights = NULL, method = "quantile", quantile = 0.5,
    kernel = "gaussian")

Arguments

- **x**: a numeric vector for which the recentered influence function is computed.
- **weights**: an optional vector of weights of x to be used in the computation of the recentered influence function. Should be NULL or a numeric vector.
- **method**: the distribution statistic for which the recentered influence function is estimated. Options are "quantile", "gini" and "variance". Default is "quantile".
- **quantile**: quantile to be used when method "quantile" is selected. Must be a numeric between 0 and 1. Default is 0.5 (median). Only a single quantile can be selected.
- **kernel**: a character giving the smoothing kernel to be used in method "quantile". Options are "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine". Default is "gaussian".

Details

The RIF can be used as input for a RIF regression approach. RIF regressions are mostly used to estimate the marginal effect of covariates on distributional statistics of income or wealth.

The RIF is calculated by adding the distributional statistic (quantile, gini or variance) to the influence function. RIF is a numeric vector where each element corresponds to a particular individual’s influence on the distributional statistic.

Value

A numeric vector of the recentered influence function of the selected distributional statistic.

References


**rifr**

### See Also

**rifr**

### Examples

```r
data(mex_inc_2008)

#Recentered influence function of 20th quantile
rif_q20 <- rif(x=mex_inc_2008$income, weights=mex_inc_2008$factor, method="quantile", quantile=0.2)

#Recentered influence function of the gini coefficient
rif_gini <- rif(x=mex_inc_2008$income, weights=mex_inc_2008$factor, method="gini")
```

---

**rifr**

Recentered influence function regression (RIF Regression)

### Description

Recentered influence function regression of a distributional statistic.

### Usage

```r
rifr(formula, data, weights = NULL, method = "quantile", quantile = 0.5, kernel = "gaussian")
```

### Arguments

- **formula**: an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted in the RIF regression.
- **data**: a data frame containing the variables and weights of the model.
- **weights**: an optional vector of weights of x to be used in the computation of the recentered influence function. Should be NULL or a numeric vector. Should be inside selected data frame in the function and between quotation marks.
- **method**: the distribution statistic for which the recentered influence function is estimated. Options are "quantile", "gini" and "variance". Default is "quantile".
- **quantile**: quantile to be used when method "quantile" is selected. Must be a numeric between 0 and 1. Default is 0.5 (median). Multiple quantiles can be used.
- **kernel**: a character giving the smoothing kernel to be used in method "quantile". Options are "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine". Default is "gaussian".
Details

RIF Regressions can be used to estimate the marginal effects of covariates on distributional statistics (such as quantiles, gini and variance). It is based on the recentered influence function of a statistic. The transformed RIF is used as the dependent variable in an ordinary least squares regression. RIF regressions are mostly used to estimate the marginal effect of covariates on distributional statistics of income or wealth.

Value

A list containing the results of the RIF regression.

- coefficients: the coefficient estimates.
- SE: the coefficient standard error.
- t: the coefficient t-value.
- p: the coefficient p-value.
- adjusted_r2: the adjusted r-squares.

References


See Also

rif, rifr, SE

Examples

data(mex_inc_2008)

#Recentered influence function of each decile
rifr_q <- rifr(income~hh_structure+education, data=mex_inc_2008, weights="factor", method="quantile", quantile=seq(0.1,0.9,0.1), kernel="gaussian")

#Recentered influence function of the gini coefficient
rifr_gini <- rifr(income~hh_structure+education, data=mex_inc_2008, weights="factor", method="gini")
Inference of recentered influence function regression (RIF regression)

Description

Inference of a RIF Regression using a bootstrap method.

Usage

```r
rifrSE(formula, data, weights = NULL, method = "quantile", quantile = 0.5,
       kernel = "gaussian", Nboot = 100, confidence = 0.95)
```

Arguments

- **formula**: an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted in the RIF regression.
- **data**: a data frame containing the variables and weights of the model.
- **weights**: an optional vector of weights of x to be used in the computation of the recentered influence function. Should be NULL or a numeric vector. Should be inside selected data frame in the function and between quotation marks.
- **method**: the distribution statistic for which the recentered influence function is estimated. Options are "quantile", "gini" and "variance". Default is "quantile".
- **quantile**: quantile to be used when method "quantile" is selected. Must be a numeric between 0 and 1. Default is 0.5 (median). Only a single quantile can be used.
- **kernel**: a character giving the smoothing kernel to be used in method "quantile". Options are "gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine" or "optcosine". Default is "gaussian".
- **Nboot**: the number of bootstrap replicates. Default is 100.
- **confidence**: significance level for estimation of the confidence interval of the fitted model. Default is 0.95.

Details

RIF Regressions can be used to estimate the marginal effects of covariates on distributional statistics (such as quantiles, gini and variance). It is based on the recentered influence function of a statistic. The transformed RIF is used as the dependent variable in an ordinary least squares regression. RIF regressions are mostly used to estimate the marginal effect of covariates on distributional statistics of income or wealth.

The standard errors, confidence intervals and Z- and P-values are calculated by using a standard bootstrap method (from boot package).
Value

A data frame containing the results of the RIF regression.

Coef    estimated coefficients of the original (non bootstrapped) RIF regression
lower   lower bound of confidence interval of estimated coefficient
upper   upper bound of confidence interval of estimated coefficient
SE      standard error
Z Value Z value
P Value P value
Signif  Significance codes of P: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

References


See Also

rif rifr

Examples

data(mex_inc_2008)

#Recentered influence function of 20th quantile
rifr.q <- rifrSE(income~hh_structure+education, data=mex_inc_2008, weights="factor", method="quantile", quantile=0.2, kernel="gaussian", Nboot=100, confidence=0.95)

#Recentered influence function of the gini coefficient
rifr.gini <- rifrSE(income~hh_structure+education, data=mex_inc_2008, weights="factor", method="gini", Nboot=100, confidence=0.95)
Description

Returns the (optional weighted) Theil index for a vector.

Usage

```r
theil.wtd(x, weights = NULL)
```

Arguments

- `x` : a numeric vector containing at least non-negative elements.
- `weights` : an optional vector of weights of `x` to be used in the computation of the Theil index. Should be `NULL` or a numeric vector.

Details

The Theil index is a measure of inequality among values of a distribution. It is a member of the Generalized Entropy Measures. Also referred to as GE(1). The index can have a value between 0 and \( \ln N \) (the logarithm of the number of values), with 0 being the lowest possible inequality. It uses a logarithmic transformation of the values of the distribution. Therefore it cannot handle negative or zero values. Those are excluded from the computation in this function. The Theil Index is more sensitive for changes in the upper tail of the distribution.

Extension of the `calcGEI` function in IC2 package in order to handle missings.

Value

The value of the Theil index.

Source


References


Examples

# calculate Theil Index using Mexican Income data set
data(mex_inc_2008)

# unweighted Theil Index:
theil.wtd(mex_inc_2008$income)

# weighted Theil Index:
theil.wtd(x=mex_inc_2008$income, weights=mex_inc_2008$factor)
Index

* datasets
  mex_inc_2008, 8
  mex_inc_2016, 10

dineq_change_rb, 2, 5
dineq_rb, 4, 4

gini.wtd, 6
gini_decomp, 7, 15

mex_inc_2008, 8
mex_inc_2016, 10
mld.wtd, 11
mld_change, 12, 15
mld_decomp, 8, 13, 14

ntiles.wtd, 15

polar.wtd, 16

rif, 18, 20, 22
rifr, 19, 19, 22
rifrSE, 20, 21

theil.wtd, 23