Package ‘Compind’

March 12, 2019

Type Package
Title Composite Indicators Functions
Version 2.1
Date 2019-3-11
Author Francesco Vidoli, Elisa Fusco
Maintainer Francesco Vidoli <fvidoli@gmail.com>
Description Contains functions to enhance approaches to the Composite Indicators methods, focusing, in particular, on the normalisation and weighting-aggregation steps.
Depends Benchmarking, psych, boot, lpSolve, spdep
Imports Hmisc, MASS, GPArotation, nonparaeff, smaa, np
License GPL-3
Suggests R.rsp
VignetteBuilder R.rsp
NeedsCompilation no
Repository CRAN
Date/Publication 2019-03-12 18:06:27 UTC

R topics documented:

Compind-package .................................................... 2
bandwidth_CI ..................................................... 3
bandwidth_CI_bad ............................................... 4
ci_ami ............................................................ 5
ci_bod ............................................................. 7
ci_bod_constr .................................................. 8
ci_bod_constr_bad ............................................. 9
ci_bod_constr_mpi ............................................ 11
ci_bod_dir ....................................................... 12
ci_bod_var_w .................................................. 14
ci_factor ......................................................... 15
ci_generalized_mean .......................................... 16
Compind package contains functions to enhance several approaches to the Composite Indicators (CIs) methods, focusing, in particular, on the normalisation and weighting-aggregation steps.

Details

Package: Compind
Type: Package
Version: 2.0
Date: 2018-1-29
Depends: Benchmarking, Hmisc, MASS, ggplot2, psych, GPArotation, lpSolve, nonparaeff, boot, np
License: GPL-3

Author(s)

Francesco Vidoli, Elisa Fusco
Maintainer: Francesco Vidoli <fvidoli@gmail.com>

References


OECD (2008) "Handbook on constructing composite indicators: methodology and user guide".


---

**bandwidth_CI**

**Multivariate mixed bandwidth selection for exogenous variables (in presence of all desirable indicators)**

**Description**

A function for the selection of optimal multivariate mixed bandwidths for the kernel density estimation of continuous and discrete exogenous variables.

**Usage**

`bandwidth_CI(x, indic_col, Q=NULL, Q_ord=NULL)`
Arguments

- **x**: A data frame containing simple indicators.
- **indic_col**: Simple indicators column number; it has to be greater than 1
- **Q**: A matrix containing continuous exogenous variables.
- **Q_ord**: A matrix containing discrete exogenous variables.

Details

Author thanks Nicky Rogge for his help and for making available the original code of the bandwidth function.

Value

- **bandwidth**: A matrix containing the optimal bandwidths for the exogenous variables indicated in Q and Q_ord.
- **ci_method**: "bandwidth_CI"

Author(s)

Fusco E., Rogge N.

Examples

```r
# Time series data on unemployment, household income and energy consumption
data(EU_2020)
indic <- c("employ_2011", "gasemiss_2011","deprived_2011")

dat <- EU_2020[-c(10,18),indic]
Q_GDP <- EU_2020[-c(10,18),"perCGDP_2011"]

# Conditional robust BoD Constr Low VWR
band = bandwidth_CI(dat, Q = Q_GDP)
```

**bandwidth_CI_bad** *Multivariate mixed bandwidth selection for exogenous variables (in presence of undesirable indicators)*

Description

A function for the selection of optimal multivariate mixed bandwidths for the kernel density estimation of continuous and discrete exogenous variables.

Usage

```r
bandwidth_CI_bad(x, indic_col, ngood, nbad, Q=NULL, Q_ord=NULL)
```
ci_ampi

Arguments

- **x**: A data frame containing simple indicators.
- **indic_col**: Simple indicators column number.
- **ngood**: The number of desirable outputs; it has to be greater than 0.
- **nbad**: The number of undesirable outputs; it has to be greater than 0.
- **Q**: A matrix containing continuous exogenous variables.
- **Q_ord**: A matrix containing discrete exogenous variables.

Details

Author thanks Nicky Rogge for his help and for making available the original code of the bandwidth function.

Value

- **bandwidth**: A matrix containing the optimal bandwidths for the exogenous variables indicated in Q and Q_ord.
- **ci_method**: "bandwidth_CI_bad"

Author(s)

Fusco E., Rogge N.

Examples

data(EU_2020)
indic <- c("employ_2011", "gasemiss_2011","deprived_2011")
dat <- EU_2020[-c(10,18),indic]
Q_GDP <- EU_2020[-c(10,18),"percGDP_2011"]

# Conditional robust BoD Constrained VWR
band = bandwidth_CI_bad(dat, ngood=1, nbad=2, Q = Q_GDP)

---

**ci_ampi**

*Adjusted Mazziotta-Pareto Index (AMPI) method*

Description

Adjusted Mazziotta-Pareto Index (AMPI) is a non-compensatory composite index that allows to take into account the time dimension, too. The calculation part is similar to the MPI framework, but the standardization part makes the scores obtained over the years comparable.

Usage

```
ci_ampi(x, indic_col, gp, time, polarity, penalty = "POS")
```
Arguments

- **x**: A data.frame containing simple indicators in a Long Data Format.
- **indic_col**: Simple indicators column number.
- **gp**: Goalposts; to facilitate the interpretation of results, the goalposts can be chosen so that 100 represents a reference value (e.g., the average in a given year).
- **time**: The time variable (mandatory); if the analysis is carried out over a single year, it is necessary to create a constant variable (i.e., `dataframe@year <- 2014`).
- **polarity**: Polarity vector: "POS" = positive, "NEG" = negative. The polarity of an individual indicator is the sign of the relationship between the indicator and the phenomenon to be measured (e.g., in a well-being index, "GDP per capita" has 'positive' polarity and "Unemployment rate" has 'negative' polarity).
- **penalty**: Penalty direction; Use "POS" (default) in case of 'increasing' or 'positive' composite index (e.g., well-being index), "NEG" in case of 'decreasing' or 'negative' composite index (e.g., poverty index).

Details

Author thanks Leonardo Alaimo for their help and for making available the original code of the AMPI function.

Value

An object of class "CI". This is a list containing the following elements:

- **ci_ampi_est**: Composite indicator estimated values.
- **ci_method**: Method used; for this function ci_method="ampi".

Author(s)

Fusco E., Alaimo L.

References


See Also

- `ci_bod`, `normalise_ci`

Examples

data(EU_2020)

data_test = EU_2020[,c("employ_2010","employ_2011","finalenergy_2010","finalenergy_2011")]

EU_2020_long<reshape(data_test, varying=c("employ_2010","employ_2011","finalenergy_2010","finalenergy_2011"),
ci_bod

direction="long",
idvar="geo",
sep="_"

CI <- ci_ampi(EU_2020_long,
   indic_col=c(2:3),
   gp=c(50, 100),
   time=EU_2020_long[,1],
   polarity= c("POS", "POS"),
   penalty="POS")
CI$ci_ampi_est

ci_bod  Benefit of the Doubt approach (BoD)

Description

Benefit of the Doubt approach (BoD) is the application of Data Envelopment Analysis (DEA) to the field of composite indicators. It was originally proposed by Melyn and Moesen (1991) to evaluate macroeconomic performance.

Usage

   ci_bod(x, indic_col)

Arguments

   x                    A data.frame containing simple indicators.
   indic_col            A numeric list indicating the positions of the simple indicators.

Value

   An object of class "CI". This is a list containing the following elements:

   ci_bod_est  Composite indicator estimated values.
   ci_method   Method used; for this function ci_method="bod".
   ci_bod_weights  Raw weights assigned to the simple indicators (Dual values - prices - in the dual DEA formulation).

Author(s)

   Vidoli F.
References

OECD (2008) "Handbook on constructing composite indicators: methodology and user guide".

See Also

ci_bod_dir, ci_rbod

Examples

```r
i1 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.2, 0.03)
i2 <- seq(0.3, 1, len = 100) - rnorm(100, 0.2, 0.03)
Indic = data.frame(i1, i2)
CI = ci_bod(Indic)
# validating BoD score
w = CI$ci_bod_weights
Indic[,]%*%w[,] + Indic[,]%*%w[,]
```

data(EU_NUTS1)
data_norm = normalise_ci(EU_NUTS1,c(2:3),polarity = c("POS","POS"), method=2)
CI = ci_bod(data_norm$ci_norm,c(1:2))

---

### ci_bod_constr

**Constrained Benefit of the Doubt approach (BoD)**

#### Description

The constrained Benefit of the Doubt function lets to introduce additional constraints to the weight variation in the optimization procedure so that all the weights obtained are greater than a lower value (low_w) and less than an upper value (up_w).

#### Usage

```r
ci_bod_constr(x, indic_col, up_w, low_w)
```

#### Arguments

- `x` : A data.frame containing simple indicators.
- `indic_col` : A numeric list indicating the positions of the simple indicators.
- `up_w` : Importance weights upper bound.
- `low_w` : Importance weights lower bound.
`ci_bod_constr_bad`

**Value**

An object of class "CI". This is a list containing the following elements:

- `ci_bod_constr_est`
  Constrained composite indicator estimated values.
- `ci_method`
  Method used; for this function `ci_method="bod_constrained"`.
- `ci_bod_constr_weights`
  Raw constrained weights assigned to the simple indicators.

**Author(s)**

Rogge N., Vidoli F.

**References**


**See Also**

`ci_bod_dir`, `ci_bod`

**Examples**

```r
i1 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.2, 0.03)
i2 <- seq(0.3, 1, len = 100) - rnorm(100, 0.2, 0.03)
Indic = data.frame(i1, i2)
CI = ci_bod_constr(Indic, up_w=1, low_w=0.05)

data(EU_NUTS1)
data_norm = normalise_ci(EU_NUTS1,c(2:3), polarity = c("POS","POS"), method=2)
CI = ci_bod_constr(data_norm$ci_norm,c(1:2),up_w=1, low_w=0.05)
```

---

**Description**

The constrained Benefit of the Doubt function introduces additional constraints to the weight variation in the optimization procedure (Constrained Virtual Weights Restriction) allowing to restrict the importance attached to a single indicator expressed in percentage terms, ranging between a lower and an upper bound (VWR); this function, furthermore, allows to calculate the composite indicator simultaneously in presence of undesirable (bad) and desirable (good) indicators allowing to impose a preference structure (ordVWR).
Usage

    ci_bod_constr_bad(x, indic_col, ngood=1, nbad=1, low_w=0, pref=NULL)

Arguments

    x                     A data.frame containing simple indicators; the order is important: first columns
                        must contain the desirable indicators, while second ones the undesirable indicators.
    indic_col             A numeric list indicating the positions of the simple indicators.
    ngood                 The number of desirable outputs; it has to be greater than 0.
    nbad                  The number of undesirable outputs; it has to be greater than 0.
    low_w                 Importance weights lower bound.
    pref                  The preference vector among indicators; For example if Indic1 is the most
                        important, Indic2,Indic3 are more important than Indic4 and no preference
                        judgment on Indic5 (= not included in the vector), the pref vector can be
                        written as: c("Indic1", "Indic2","Indic3","Indic4")

Value

    An object of class "CI". This is a list containing the following elements:

    ci_bod_constr_bad_est
                        Composite indicator estimated values.
    ci_method            Method used; for this function ci_method="bod_constr_bad".
    ci_bod_constr_bad_weights
                        Raw weights assigned to each simple indicator.
    ci_bod_constr_bad_target
                        Indicator target values.

Author(s)

    Fusco E., Rogge N.

References

    Rogge N., de Jaeger S. and Lavigne C. (2017) "Waste Performance of NUTS 2-regions in the EU: A
    Conditional Directional Distance Benefit-of-the-Doubt Model", Ecological Economics, vol.139,
    pp. 19-32.

    Zanella A., Camanho A.S. and Dias T.G. (2015) "Undesirable outputs and weighting schemes in
    composite indicators based on data envelopment analysis", European Journal of Operational Re-

See Also

    ci_bod_constr
Examples

data(EU_2020)
dat <- EU_2020[-c(10,18),indic]

# BoD Constrained VWR
CI_BoD_C = ci_bod_constr_bad(dat, ngood=2, nbad=2, low_w=0.05, pref=NULL)
CI_BoD_C$ci_bod_constr_bad_est

# BoD Constrained ordVWR
importance <- c("gasemiss_2011","percGDP_2011","employ_2011")
CI_BoD_C = ci_bod_constr_bad(dat, ngood=2, nbad=2, low_w=0.05, pref=importance)
CI_BoD_C$ci_bod_constr_bad_est

---

**ci_bod_constr_mpi**  
*Non Compensative Constrained Benefit of the Doubt approach (BoD)*

Description

The constrained Benefit of the Doubt function lets to introduce additional constraints to the weight variation in the optimization procedure so that all the weights obtained are greater than a lower value (low_w) and less than an upper value (up_w). In a second step the composite indicator is adjusted by a 'penalty' coefficient related to the variability of each unit (see, method of the coefficient of variation penalty - ci_mpi)

Usage

`ci_bod_constr_mpi(x,indic_col,up_w,low_w,penalty="POS")`

Arguments

- **x**: A data.frame containing simple indicators.
- **indic_col**: A numeric list indicating the positions of the simple indicators.
- **up_w**: Importance weights upper bound.
- **low_w**: Importance weights lower bound.
- **penalty**: Penalty direction; Use "POS" (default) in case of 'increasing' or 'positive' composite index (e.g., well-being index), "NEG" in case of 'decreasing' or 'negative' composite index (e.g., poverty index).

Value

An object of class "CI". This is a list containing the following elements:

- **ci_bod_constr_est_mpi**: Constrained composite indicator estimated values.
- **ci_bod_constr mpi_pen**: Penalized constrained composite indicator estimated values.
ci_bod_constr_mpi_weights
Raw constrained weights (not penalized) assigned to the simple indicators.

ci_method
Method used; for this function ci_method="bod_constrained_mpi".

Author(s)
Vidoli F.

References


See Also
ci_bod_constr, ci_mpi

Examples

data(EU_NUTS1)
data_norm = normalise_ci(EU_NUTS1,
c(2:3),
c("NEG","POS"),
method=1,
z.mean=100,
z.std=10)

CI = ci_bod_constr_mpi(data_norm$ci_norm,
c(1:2),
up_w=1,
low_w=0.1,
penalty="POS")

---

**ci_bod_dir**  
*Directional Benefit of the Doubt (D-BoD) model*

Description
Directional Benefit of the Doubt (D-BoD) model enhance non-compensatory property by introducing directional penalties in a standard BoD model in order to consider the preference structure among simple indicators.
ci_bod_dir

Usage

    ci_bod_dir(x, indic_col, dir)

Arguments

  x         A data.frame containing score of the simple indicators.
  indic_col Simple indicators column number.
  dir       Main direction. For example you can set the average rates of substitution.

Value

  An object of class "CI". This is a list containing the following elements:

  ci_bod_dir_est Composite indicator estimated values.
  ci_method      Method used; for this function ci_method="bod_dir".

Author(s)

  Vidoli F., Fusco E.

References


See Also

  ci_bod, ci_rbod

Examples

    i1 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.2, 0.03)
    i2 <- seq(0.3, 1, len = 100) - rnorm(100, 0.2, 0.03)
    Indic = data.frame(i1, i2)
    CI = ci_bod_dir(Indic, dir=c(1,1))

    data(EU_NUTS1)
    data_norm = normalise_ci(EU_NUTS1,c(2:3),polarity = c("POS","POS"), method=2)
    CI = ci_bod_dir(data_norm$ci_norm,c(1:2),dir=c(1,0.5))
Description

Variance weighted Benefit of the Doubt approach (BoD variance weighted) is a particular form of BoD method with additional information in the optimization problem. In particular it has been added weight constraints (in form of an Assurance region type I (AR I)) endogenously determined in order to take into account the ratio of the vertical variability of each simple indicator relative to one another.

Usage

ci_bod_var_w(x, indic_col, boot_rep = 5000)

Arguments

x A data.frame containing score of the simple indicators.
indic_col Simple indicators column number.
boot_rep The number of bootstrap replicates (default=5000) for the estimates of the non-parametric bootstrap (first order normal approximation) confidence intervals for the variances of the simple indicators.

Details

For more informations about the estimation of the confidence interval for the variances, please see function boot.ci, package boot.

Value

An object of class "CI". This is a list containing the following elements:

ci_bod_var_w_est Composite indicator estimated values.
ci_method Method used; for this function ci_method="bod_var_w".

Author(s)

Vidoli F.

References

ci_factor

Weighting method based on Factor Analysis

Description

Factor analysis groups together collinear simple indicators to estimate a composite indicator that captures as much as possible of the information common to individual indicators.

Usage

```
ci_factor(x, indic_col, method = "ONE", dim)
```

Arguments

- **x**: A data.frame containing score of the simple indicators.
- **indic_col**: Simple indicators column number.
- **method**: If method = "ONE" (default) the composite indicator estimated values are equal to first component scores; if method = "ALL" the composite indicator estimated values are equal to component score multiplied by its proportion variance; if method = "CH" it can be choose the number of the component to take into account.
- **dim**: Number of chosen component (if method = "CH", default is 3).

Value

An object of class "CI". This is a list containing the following elements:

- **ci_factor_est**: Composite indicator estimated values.
- **loadings_fact**: Variance explained by principal factors (in percentage terms).
- **ci_method**: Method used; for this function ci_method="factor".

Author(s)

Vidoli F.
References

OECD (2008) "Handbook on constructing composite indicators: methodology and user guide".

See Also

ci_bod, ci_mpi

Examples

i1 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.2, 0.03)
i2 <- seq(0.3, 1, len = 100) - rnorm(100, 0.2, 0.03)
Indic = data.frame(i1, i2)
CI = ci_factor(Indic)

data(EU_NUTS1)
CI = ci_factor(EU_NUTS1[,2:3], method=“ALL”)

data(EU_2020)
data_norm = normalise_ci(EU_2020[,47:51],polarity = c("POS","POS","POS","POS","POS"), method=2)
CI3 = ci_factor(data_norm$ci_norm[,1:5],method="CH", dim=3)

---

**ci_generalized_mean**  
*Weighting method based on generalized mean*

Description

Generalized means are a family of functions for aggregating sets of numbers (it include as special cases the Pythagorean means, arithmetic, geometric, and harmonic means). The generalized mean is also known as power mean or Holder mean.

Usage

ci_generalized_mean(x, indic_col, p, na.rm=TRUE)

Arguments

- **x**  
  A data.frame containing simple indicators.
- **indic_col**  
  Simple indicators column number.
- **p**  
  Exponent p (real number).
- **na.rm**  
  Remove NA values before processing; default is TRUE.

Value

An object of class "CI". This is a list containing the following elements:

- **ci_generalized_mean_est**  
  Composite indicator estimated values.
- **ci_method**  
  Method used; for this function ci_method="generalized_mean".
Note

The generalized mean with the exponent $p$ can be expressed as:

$$M_p(I_1, \ldots, I_n) = \left( \frac{1}{n} \sum_{i=1}^{n} I_i^p \right)^{\frac{1}{p}}$$

Particular cases are: $p = -\infty$: minimum, $p = -1$: harmonic mean, $p = 0$: geometric mean, $p = 1$: arithmetic mean, $p = 2$: root-mean-square and $p = \infty$: maximum.

Author(s)

Vidoli F.

See Also

`ci_geom_gen, ci_factor`

Examples

```r
i1 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.2, 0.03)
i2 <- seq(0.3, 1, len = 100) - rnorm(100, 0.2, 0.03)
Indic = data.frame(i1, i2)
CI = ci_generalized_mean(Indic, p=-1) # harmonic mean
data(EU_NUTS1)
CI = ci_generalized_mean(EU_NUTS1,c(2:3),p=2) # geometric mean
```

---

**ci_geom_bod_intertemp**  Intertemporal analysis for geometric mean quantity index numbers

Description

Intertemporal analysis for geometric mean quantity index numbers with Benefit-of-the-Doubt weights - see function `ci_bod_constr`.

Usage

```r
ci_geom_bod_intertemp(x0,x1,indic_col,up_w,low_w,bench)
```

Arguments

- **x0**: A data.frame containing simple indicators - time 0
- **x1**: A data.frame containing simple indicators - time 1
- **indic_col**: A numeric list indicating the positions of the simple indicators.
- **up_w**: Weights upper bound.
- **low_w**: Weights lower bound.
- **bench**: Row number of the benchmark unit
Value

An object of class "CI". This is a list containing the following elements:

- `ci_geom_bod_intertemp_est`: A matrix containing the Overall Change (period t1 vs t0), the Change Effect (period t1 vs t0), the Benchmark Effect (period t1 vs t0) and Weight Effect (period t1 vs t0).

- `ci_method`: Method used; for this function ci_method="Intertemporal_effects_Geometric_BoD".

Author(s)

Rogge N., Vidoli F.

References


See Also

ci_bod_constr, ci_bod

Examples

```r
i1_t1 <- seq(0.3, 0.5, len = 100)
i2_t1 <- seq(0.3, 1, len = 100)
Indic_t1 = data.frame(i1_t1, i2_t1)

i1_t0 <- i1_t1 - rnorm(100, 0.2, 0.03)
i2_t0 <- i2_t1 - rnorm(100, 0.2, 0.03)
Indic_t0 = data.frame(i1_t0, i2_t0)

intertemp = ci_geom_bod_intertemp(Indic_t0, Indic_t1, c(1:2), up_w=0.95, low_w=0.05, 1)
intertemp
```

---

`ci_geom_gen`  
**Generalized geometric mean quantity index numbers**

Description

This function use the geometric mean to aggregate the single indicators. Two weighting criteria has been implemented: EQUAL: equal weighting and BOD: Benefit-of-the-Doubt weights following the Puyenbroeck and Rogge (2017) approach.
Usage

```r
ci_geom_gen(x, indic_col, meth, up_w, low_w, bench)
```

Arguments

- `x`: A data.frame containing simple indicators.
- `indic_col`: A numeric list indicating the positions of the simple indicators.
- `up_w`: if meth="BOD"; upper bound of the weighting set.
- `low_w`: if meth="BOD"; lower bound of the weighting set.
- `bench`: Row number of the benchmark unit used to normalize the data.frame `x`.

Value

An object of class "CI". This is a list containing the following elements:

If `meth` = "EQUAL":

- `ci_mean_geom_est`: Composite indicator estimated values.
- `ci_method`: Method used; for this function `ci_method="mean_geom"`.

If `meth` = "BOD":

- `ci_geom_bod_est`: Constrained composite indicator estimated values.
- `ci_geom_bod_weights`: Raw constrained weights assigned to the simple indicators.
- `ci_method`: Method used; for this function `ci_method="geometric_bod"`.

Author(s)

Rogge N., Vidoli F.

References


See Also

`ci_bod_dir, ci_bod`
ci_mean_min

Examples

```r
i1 <- seq(0.3, 1, len = 100) - rnorm(100, 0.1, 0.03)
i2 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.1, 0.03)
i3 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.1, 0.03)
Indic = data.frame(i1, i2, i3)

geom1 = ci.geom.gen(Indic, c(1:3), meth = "EQUAL")
geom1$ci.mean.geom.est
geom1$ci_method

geom2 = ci.geom.gen(Indic, c(1:3), meth = "BOD", 0.7, 0.3, 100)
geom2$ci.geom.bod.est
geom2$ci.geom.bod.weights
```

Description

The Mean-Min Function (MMF) is an intermediate case between arithmetic mean, according to which no unbalance is penalized, and min function, according to which the penalization is maximum. It depends on two parameters that are respectively related to the intensity of penalization of unbalance ($\alpha$) and intensity of complementarity ($\beta$) among indicators.

Usage

```r
ci_mean_min(x, indic_col, alpha, beta)
```

Arguments

- `x`: A data.frame containing simple indicators.
- `indic_col`: Simple indicators column number.
- `alpha`: The intensity of penalisation of unbalance among indicators, $0 \leq \alpha \leq 1$
- `beta`: The intensity of complementarity among indicators, $\beta \geq 0$

Value

An object of class "CI". This is a list containing the following elements:

- `ci_mean_min_est`: Composite indicator estimated values.
- `ci_method`: Method used; for this function `ci_method="mean_min"`.
**ci_mpi**

**Author(s)**

Vidoli F.

**References**


**See Also**

*ci_mpi*, *normalise_ci*

**Examples**

```r
data(EU_NUTS1)
data_norm = normalise_ci(EU_NUTS1,c(2:3),c("NEG","POS"),method=2)
CI = ci_mean_min(data_norm$ci_norm, alpha=0.5, beta=1)
```

---

**ci_mpi**

*Mazziotta-Pareto Index (MPI) method*

**Description**

Mazziotta-Pareto Index (MPI) is a non-linear composite index method which transforms a set of individual indicators in standardized variables and summarizes them using an arithmetic mean adjusted by a "penalty" coefficient related to the variability of each unit (method of the coefficient of variation penalty).

**Usage**

`ci_mpi(x, indic_col, penalty="POS")`

**Arguments**

- **x**: A data.frame containing simple indicators.
- **indic_col**: Simple indicators column number.
- **penalty**: Penalty direction; Use "POS" (default) in case of 'increasing' or 'positive' composite index (e.g., well-being index), "NEG" in case of 'decreasing' or 'negative' composite index (e.g., poverty index).

**Value**

An object of class "CI". This is a list containing the following elements:

- **ci_mpi_est**: Composite indicator estimated values.
- **ci_method**: Method used; for this function ci_method="mpi".
Author(s)
Vidoli F.

References

See Also
ci_bod, normalise_ci

Examples
data(EU_NUTS1)

# Please, pay attention. MPI can be calculated only with two standardizations methods:
# Classic MPI - method=1, z.mean=100 and z.std=10
# Correct MPI - method=2
# For more info, please see references.
data_norm = normalise_ci(EU_NUTS1,c(2:3),c("NEG","POS"),method=1,z.mean=100, z.std=10)
CI = ci_mpi(data_norm$ci_norm, penalty="NEG")

data(EU_NUTS1)
CI = ci_mpi(EU_NUTS1,c(2:3),penalty="NEG")

---

**ci_rbod**

Robust Benefit of the Doubt approach (RBoD)

Description
Robust Benefit of the Doubt approach (RBoD) is the robust version of the BoD method. It is based on the concept of the expected minimum input function of order-\(m\) so "in place of looking for the lower boundary of the support of \(F\), as was typically the case for the full-frontier (DEA or FDH), the order-\(m\) efficiency score can be viewed as the expectation of the maximal score, when compared to \(m\) units randomly drawn from the population of units presenting a greater level of simple indicators", Daraio and Simar (2005).

Usage
ci_rbod(x,indic_col,M,B)

Arguments

- \(x\) A data.frame containing score of the simple indicators.
- \(\text{indic\_col}\) Simple indicators column number.
- \(M\) The number of elements in each of the bootstrapped samples.
- \(B\) The number of bootstrap replicates.
Value

An object of class "CI". This is a list containing the following elements:

- \texttt{ci\_rbod\_est} Composite indicator estimated values.
- \texttt{ci\_method} Method used; for this function \texttt{ci\_method}="rbod".

Author(s)

Vidoli F.

References


See Also

ci\_bod

Examples

```r
i1 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.2, 0.03)
i2 <- seq(0.3, 1, len = 100) - rnorm(100, 0.2, 0.03)
Indic = data.frame(i1, i2)
CI = ci\_rbod(Indic, B=10)

data(EU\_NUTS1)
data\_norm = normalise\_ci(EU\_NUTS1,c(2:3), polarity = c("POS","POS"), method=2)
CI = ci\_rbod(data\_norm\_ci\_norm,c(1:2),M=10,B=20)
```

\textbf{ci\_rbod\_constr\_bad} \hspace{1cm} Robust constrained Benefit of the Doubt approach (BoD) in presence of undesirable indicators

Description

The Robust constrained Benefit of the Doubt function introduces additional constraints to the weight variation in the optimization procedure (Constrained Virtual Weights Restriction) allowing to restrict the importance attached to a single indicator expressed in percentage terms, ranging between a lower and an upper bound (VWR); this function, furthermore, allows to calculate the composite indicator simultaneously in presence of undesirable (bad) and desirable (good) indicators allowing to impose a preference structure (ordVWR). This function is the robust version of the \texttt{ci\_bod\_constr\_bad}; it is based on the concept of the expected minimum input function of order-$m$ (Daraio and Simar, 2005) allowing to compare the unit under analysis against $M$ peers by extracting $B$ samples with replacement.
**Usage**

```r
ci_rbod_constr_bad(x, indic_col, ngood=1, nbad=1, low_w=0, pref=NULL, M, B)
```

**Arguments**

- **x**: A data.frame containing simple indicators.
- **indic_col**: A numeric list indicating the positions of the simple indicators.
- **ngood**: The number of desirable outputs; it has to be greater than 0.
- **nbad**: The number of undesirable outputs; it has to be greater than 0.
- **low_w**: Importance weights lower bound.
- **pref**: The preference vector among indicators; For example if Indic1 is the most important, Indic2, Indic3 are more important than Indic4 and no preference judgment on Indic5 (= not included in the vector), the `pref` vector can be written as: `c("Indic1", "Indic2","Indic3","Indic4")`
- **M**: The number of elements in each of the bootstrapped samples.
- **B**: The number of bootstrap replicates.

**Value**

An object of class "CI". This is a list containing the following elements:

- **ci_rbod_constr_bad_est**: Composite indicator estimated values.
- **ci_method**: Method used; for this function `ci_method="rbod_constr_bad"`.
- **ci_rbod_constr_bad_weights**: Raw weights assigned to each simple indicator.
- **ci_rbod_constr_bad_target**: Indicator target values.

**Author(s)**

Fusco E., Rogge N.

**References**


**See Also**

`ci_bod_constr`, `ci_bod_constr_bad`
Examples

data(EU_2020)
dat <- EU_2020[-c(10,18),indic]

# Robust BoD Constrained VWR
CI_BoD_C = ci_rbod_constr_bad(dat, ngood=2, nbad=2, low_w=0.05, pref=NULL, M=10, B=50)
CI_BoD_C$c$ci_rbod_constr_bad_est

# Robust BoD Constrained ordVWR
#importance <- c("gasemiss_2011","percGDP_2011","employ_2011")
#CI_BoD_C = ci_rbod_constr_bad(dat, ngood=2, nbad=2, low_w=0.05, pref=importance, M=10, B=50)
#CI_BoD_C$c$ci_rbod_constr_bad_est

---

**ci_rbod_constr_bad_Q**  
Conditional robust constrained Benefit of the Doubt approach (BoD)  
in presence of undesirable indicators

---

Description

The Conditional robust constrained Benefit of the Doubt function introduces additional constraints to the weight variation in the optimization procedure (Constrained Virtual Weights Restriction) allowing to restrict the importance attached to a single indicator expressed in percentage terms, ranging between a lower and an upper bound (VWR); this function, furthermore, allows to calculate the composite indicator simultaneously in presence of undesirable (bad) and desirable (good) indicators allowing to impose a preference structure (ordVWR). This function, in addition to being robust against outlier data (see `ci_rbod_constr_bad` function) allows to take into account external contextual continuous (Q) or/and ordinal (Q_ord) variables.

Usage

```r
ci_rbod_constr_bad_Q(x, indic_col, ngood=1, nbad=1, low_w=0, pref=NULL, M, B, Q=NULL, Q Ord=NULL, bandwidth)
```

Arguments

- `x`: A data.frame containing simple indicators.
- `indic_col`: A numeric list indicating the positions of the simple indicators.
- `ngood`: The number of desirable outputs; it has to be greater than 0.
- `nbad`: The number of undesirable outputs; it has to be greater than 0.
- `low_w`: Importance weights lower bound.
- `pref`: The preference vector among indicators; For example if Indic1 is the most important, Indic2,Indic3 are more important than Indic4 and no preference judgment on Indic5 (= not included in the vector), the `pref` vector can be written as: `c("Indic1", "Indic2","Indic3","Indic4")`
- `M`: The number of elements in each of the bootstrapped samples.
ci_rbod_constr_bad_Q

<table>
<thead>
<tr>
<th>B</th>
<th>The number of bootstrap replicates.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>A matrix containing continuous exogenous variables.</td>
</tr>
<tr>
<td>Q_ord</td>
<td>A matrix containing discrete exogenous variables.</td>
</tr>
<tr>
<td>bandwidth</td>
<td>Multivariate mixed bandwidth for exogenous variables; it can be calculated by bandwidth_CI function.</td>
</tr>
</tbody>
</table>

**Value**

An object of class "CI". This is a list containing the following elements:

- `ci_rbod_constr_bad_Q_est`: Composite indicator estimated values.
- `ci_method`: Method used; for this function `ci_method="rbod_constr_bad_Q"`.
- `ci_rbod_constr_bad_Q_weights`: Raw weights assigned to each simple indicator.
- `ci_rbod_constr_bad_Q_target`: Indicator target values.

**Author(s)**

Fusco E., Rogge N.

**References**


**See Also**

`ci_rbod_constr_bad`, `ci_bod_constr_bad`

**Examples**

```r
data(EU_2020)

indic <- c("employ_2011", "gasemiss_2011","deprived_2011")
dat <- EU_2020[-c(10,18),indic]
Q_GDP <- EU_2020[-c(10,18),"percGDP_2011"]

# Conditional robust BoD Constrained VWR
band = bandwidth_CI_bad(dat, ngood=1, nbad=2, Q = Q_GDP)

CI_BoD_C = ci_rbod_constr_bad_Q(dat, ngood=1, nbad=2)
```
ci_rbod_constr_Q

low_w=0.05,
pref=NULL,
M=10,
B=50,
Q=Q_GDP,
bandwidth = band$bandwidth)

CI_BoD_C$ci_rbod_constr_bad_Q_est

# Conditional robust BoD Constrained ordVWR
# import <- c("gasemiss_2011","employ_2011", "deprived_2011")
#
# CI_BoD_C2 = ci_rbod_constr_bad_Q(dat,
#     ngood=1,
#     nbad=2,
#     low_w=0.05,
#     pref=import,
#     M=10,
#     B=50,
#     Q=Q_GDP,
#     bandwidth = band$bandwidth)
# CI_BoD_C2$ci_rbod_constr_bad_Q_est

---

**ci_rbod_constr_Q**  
*Conditional robust constrained Benefit of the Doubt approach (BoD)*

**Description**

The Conditional robust constrained Benefit of the Doubt function introduces additional constraints to the weight variation in the optimization procedure (Constrained Virtual Weights Restriction) allowing to restrict the importance attached to a single indicator expressed in percentage terms, ranging between a lower and an upper bound (VWR); this function, furthermore, allows to calculate the composite indicator allowing to impose a preference structure (ordVWR). This function, in addition to being robust against outlier data (see `ci_rbod_constr_bad` function) allows to take into account external contextual continuous (Q) or ordinal (Q_ord) variables.

**Usage**

```r
ci_rbod_constr_Q(x, indic_col, 
low_w=0, pref=NULL, M, B, Q=NULL, Q_ord=NULL, bandwidth)
```

**Arguments**

- `x`  
  A data.frame containing simple indicators.
- `indic_col`  
  A numeric list indicating the positions of the simple indicators.
- `low_w`  
  Importance weights lower bound.
The preference vector among indicators; For example if Indicator1 is the most important, Indicator2, Indicator3 are more important than Indicator4 and no preference judgment on Indicator5 (= not included in the vector), the `pref` vector can be written as: `c("Indicator1", "Indicator2","Indicator3","Indicator4")`

M The number of elements in each of the bootstrapped samples.
B The number of bootstrap replicates.
Q A matrix containing continuous exogenous variables.
Q_ord A matrix containing discrete exogenous variables.
bandwidth Multivariate mixed bandwidth for exogenous variables; it can be calculated by `bandwidth_CI` function.

Value

An object of class "CI". This is a list containing the following elements:

- `ci_rbod_constr_Q_est`: Composite indicator estimated values.
- `ci_method`: Method used; for this function `ci_method="rbod_constr_Q"`. 
- `ci_rbod_constr_Q_weights`: Raw weights assigned to each simple indicator.
- `ci_rbod_constr_Q_target`: Indicator target values.

Author(s)

Fusco E., Rogge N., Vidoli F.

References


See Also

`ci_rbod_constr_bad`, `ci_bod_constr_bad`

Examples

data(EU_2020)

indic <- c("employ_2011", "gasemiss_2011","deprived_2011")
dat <- EU_2020[-c(10,18),indic]
Q_GDP <- EU_2020[-c(10,18),"percGDP_2011"]
Directional Robust Benefit of the Doubt approach (D-RBoD)

Description
Directional Robust Benefit of the Doubt approach (D-RBoD) is the directional robust version of the BoD method.

Usage
`ci_rbod_dir(x, indic_col, M, B, dir)`

Arguments
- `x` A data.frame containing score of the simple indicators.
- `indic_col` Simple indicators column number.
- `M` The number of elements in each of the bootstrapped samples.
- `B` The number of bootstrap replicates.
- `dir` Main direction. For example you can set the average rates of substitution.

Value
An object of class "CI". This is a list containing the following elements:
- `ci_rbod_dir_est` Composite indicator estimated values.
- `ci_method` Method used; for this function `ci_method="rbod_dir"`.

Author(s)
Fusco E., Vidoli F.
References


See Also

ci_bod, ci_rbod

Examples

data(EU_NUTS1)
data_norm = normalise_ci(EU_NUTS1,c(2:3),polarity = c("POS","POS"), method=2)
CI = ci_rbod_dir(data_norm$ci_norm, c(1:2), M = 25, B = 50, c(1,0.1))

<table>
<thead>
<tr>
<th>ci_rbod_spatial</th>
<th>Spatial robust Benefit of the Doubt approach (Sp-RBoD)</th>
</tr>
</thead>
</table>

Description

The Spatial robust Benefit of the Doubt approach (Sp-RBoD) method allows to take into account the spatial contextual condition into the robust Benefit of the Doubt method.

Usage

ci_rbod_spatial(x, indic_col, M=20, B=100, W)

Arguments

x A data.frame containing score of the simple indicators.
indic_col Simple indicators column number.
M The number of elements in each of the bootstrapped samples; default is 20.
B The number of bootstrap replicates; default is 100.
W The spatial weights matrix. A square non-negative matrix with no NAs representing spatial weights; may be a matrix of class "sparseMatrix" (spdep package)

Value

An object of class "CI". This is a list containing the following elements:

ci_rbod_spatial_est Composite indicator estimated values.
ci_method Method used; for this function ci_method="rbod_spatial".
ci_smaa_constr

Author(s)

Fusco E., Vidoli F.

References


See Also

ci_rbod

Examples

data(EU_NUTS1)

coord = EU_NUTS1[,c("Long","Lat")]
k<-knearnigh(as.matrix(coord), k=5)
k_nb<-knn2nb(k)
W_mat <-nb2mat(k_nb,style="W",zero.policy=TRUE)

CI = ci_rbod_spatial(EU_NUTS1,c(2:3),M=10,B=20, W=W_mat)

---

ci_smaa_constr  Constrained stochastic multi-objective acceptability analysis (C-SMAA)

Description

Stochastic multiobjective acceptability analysis (SMAA) is a multicriteria decision support technique for multiple decision makers based on exploring the weight space. Inaccurate or uncertain input data can be represented as probability distributions. In SMAA the decision makers need not express their preferences explicitly or implicitly; instead the technique analyses what kind of valuations would make each alternative the preferred one. The method produces for each alternative an acceptability index measuring the variety of different valuations that support that alternative, a central weight vector representing the typical valuations resulting in that decision, and a confidence factor measuring whether the input data is accurate enough for making an informed decision. (R Lahdelma, J. Hokkanen and P. Salminen, 1998); this function, in particular, allows to restricts the range of allowable weights within the SMAA analysis.

Usage

ci_smaa_constr(x,indic_col,rep, label, low_w=NULL)
Arguments

- x: A data.frame containing simple indicators.
- indic_col: A numeric list indicating the positions of the simple indicators.
- rep: Number of samples.
- label: A factor column useful to identify units.
- low_w: Importance weights lower bound vector; default is NULL (for standard SMAA).

Details

Author thanks Giuliano Resce and Raffaele Lagravinese for their help and for making available the original code of the SMAA function. The lower bound vector must be set as a vector of the same size as the number of simple indicators; for example - in the presence of two indicators - if you want to constrain only one indicator, you must write: low_w = c(0,0).

Value

An object of class "CI". This is a list containing the following elements:

- ci_smaa_constr_rank_freq: Frequence of the SMAA ranks based on the sampled alternatives’ values. The rows represent the analysis units while the first column represents the number of times the unit was in first rank, the second one in second rank and so on.
- ci_smaa_constr_average_rank: The average rank.
- ci_smaa_constr_values: The alternative values based on a set of samples from the criteria values distribution and the samples set from the feasible weight space.
- ci_method: Method used; for this function ci_method="smaa_const".

Author(s)

Vidoli F.

References

- S. Greco, A. Ishizaka, B. Matarazzo and G. Torrisi (2017) "Stochastic multi-attribute acceptability analysis (SMAA): an application to the ranking of Italian regions", Regional Studies

See Also

- ci_bod
Description

Wroclaw taxonomy method (also known as the dendric method), originally developed at the University of Wroclaw, is based on the distance from a theoretical unit characterized by the best performance for all indicators considered; the composite indicator is therefore based on the sum of euclidean distances from the ideal unit and normalized by a measure of variability of these distance (mean + 2*std).

Usage

ci_wroclaw(x, indic_col)

Arguments

x A data.frame containing simple indicators.
indic_col Simple indicators column number.

Details

Please pay attention that ci_wroclaw_est is the distance from the "ideal" unit; so, units with higher values for the simple indicators get lower values of composite indicator.

Value

An object of class "CI". This is a list containing the following elements:

ci_wroclaw_est Composite indicator estimated values.
ci_method Method used; for this function ci_method="wroclaw".
Author(s)

Vidoli F.

References


See Also

ci_bod, ci_mpi

Examples

```r
i1 <- seq(0.3, 0.5, len = 100) - rnorm(100, 0.2, 0.03)
i2 <- seq(0.3, 1, len = 100) - rnorm(100, 0.2, 0.03)
Indic = data.frame(i1, i2)
CI = ci_wroclaw(Indic)

data(EU_NUTS1)
CI = ci_wroclaw(EU_NUTS1,c(2:3))

data(EU_2020)
data_selez = EU_2020[.c(1,22,191)]
data_norm = normalise_ci(data_selez,c(2:3),c("POS","NEG"),method=3)
ci_wroclaw(data_norm$ci_norm,c(1:2))
```

EU_2020  Europe 2020 indicators

Description

Europe 2020, a strategy for jobs and smart, sustainable and inclusive growth, is based on five EU headline targets which are currently measured by eight headline indicators, Headline indicators, Eurostat, year 1990-2012 (Last update: 21/11/2013).


Usage

data(EU_2020)
EU_2020 is a dataset with 30 observations and 12 indicators (190 indicator per year).

**geo** EU-Member States including EU (28 countries) and EU (27 countries) row.

**employ** Employment rate - age group 20-64, year XXXX (1992-2012).

**perc_GDP** Gross domestic expenditure on R&D (GERD), year XXXX (1990-2012).


**share_ren** Share of renewable energy in gross final energy consumption, year XXXX (2004-2011).

**prim_ener** Primary energy consumption, year XXXX (1990-2011).

**final_energy** Final energy consumption, year XXXX (1990-2011).

**final_ener** Early leavers from education and training - Perc. of the population aged 18-24 with at most lower secondary education and not in further education or training, year XXXX (1992-2012).

**tertiary** Tertiary educational attainment - age group 30-34, year XXXX (2000-2012).

**risk_poverty** People at risk of poverty or social exclusion - 1000 persons Perc. of total population, year XXXX (2004-2012).

**low_work** People living in households with very low work intensity - 1000 persons Perc. of total population, year XXXX (2004-2012).

**risk_poverty** People at risk of poverty after social transfers - 1000 persons Perc. of total population, year XXXX (2003-2012).

**deprived** Severely materially deprived people - 1000 persons Perc. of total population, year XXXX (2003-2012).

**Author(s)**

Vidoli F.

**References**

https://ec.europa.eu/info/strategy/european-semester/framework/europe-2020-strategy_en

**Examples**

`data(EU_2020)`
**EU_NUTS1**

**EU NUTS1 Transportation data**

### Description

Eurostat regional transport statistics (reg_tran) data, year 2012.

### Usage

```r
data(EU_NUTS1)
```

### Format

EU_NUTS1 is a dataset with 34 observations and two indicators describing transportation infrastructure endowment of the main (in terms of population and GDP) European NUTS1 regions: France, Germany, Italy, Spain (United Kingdom has been omitted, due to lack of data concerning railways).

- **roads**: Calculated as \((2 \times \text{Motorways} - \text{Kilometres per 1000 km}^2 + \text{Other roads} - \text{Kilometres per 1000 km}^2)/3\)

- **trains**: Calculated as \((2 \times \text{Railway lines double} + \text{Electrified railway lines})/3\)

### Author(s)

Vidoli F.

### References


### Examples

```r
data(EU_NUTS1)
```

---

**normalise_ci**

**Normalisation and polarity functions**

### Description

This function lets to normalise simple indicators according to the polarity of each one.

### Usage

```r
normalise_ci(x, indic_col, polarity, method=1, z.mean=0, z.std=1, ties.method ="average")
```
**Arguments**

- **x**: A data frame containing simple indicators.
- **indic_col**: Simple indicators column number.
- **method**: Normalisation methods:
  - **1** (default) = standardization or z-scores using the following formulation:
    \[
    z_{ij} = z.mean \pm \frac{x_{ij} - M_{x_j}}{S_{x_j}} \cdot z.std
    \]
    where \(\pm\) depends on `polarity` parameter and `z.mean` and `z.std` represent the shifting parameters.
  - **2** = Min-max method using the following formulation:
    if `polarity`="POS":
    \[
    \frac{x - \min(x)}{\max(x) - \min(x)}
    \]
    if `polarity`="NEG":
    \[
    \frac{\max(x) - x}{\max(x) - \min(x)}
    \]
  - **3** = Ranking method. If `polarity`="POS" ranking is increasing, while if `polarity`="NEG" ranking is decreasing.
- **polarity**: Polarity vector: "POS" = positive, "NEG" = negative. The polarity of an individual indicator is the sign of the relationship between the indicator and the phenomenon to be measured (e.g., in a well-being index, "GDP per capita" has ‘positive’ polarity and "Unemployment rate" has ‘negative’ polarity).
- **z.mean**: If `method`=1, Average shifting parameter. Default is 0.
- **z.std**: If `method`=1, Standard deviation expansion parameter. Default is 1.
- **ties.method**: If `method`=3, A character string specifying how ties are treated, see `rank` for details. Default is "average".

**Value**

- **ci_norm**: A data frame containing normalised score of the chosen simple indicators.
- **norm_method**: Normalisation method used.

**Author(s)**

Vidoli F.

**References**


**See Also**

`ci_bod, ci_mpi`
Examples

data(EU_NUTS1)

# Standard z-scores normalisation #
data_norm = normalise_ci(EU_NUTS1[,c(2:3)],c("NEG","POS"),method=1,z.mean=0, z.std=1)
summary(data_norm$ci_norm)

# Normalisation for MPI index #
data_norm = normalise_ci(EU_NUTS1[,c(2:3)],c("NEG","POS"),method=1,z.mean=100, z.std=10)
summary(data_norm$ci_norm)

data_norm = normalise_ci(EU_NUTS1[,c(2:3)],c("NEG","POS"),method=2)
summary(data_norm$ci_norm)
Index

bandwidth_CI, 3
bandwidth_CI_bad, 4

ci_ami, 5
ci_bod, 6, 7, 9, 13, 15, 16, 18, 19, 22, 23, 30, 32, 34, 37
ci_bod_constr, 8, 10, 12, 17, 18, 24
ci_bod_constr_bad, 9, 24, 26, 28
ci_bod_constr_mpi, 11
ci_bod_dir, 8, 9, 12, 19
ci_bod_var_w, 14
ci_factor, 15, 17
ci_generalized_mean, 16
ci_geom_bod_intertemp, 17
ci_geom_gen, 17, 18
ci_mean_min, 20
ci_mpi, 12, 16, 21, 21, 34, 37
ci_rbod, 8, 13, 15, 22, 30, 31
ci_rbod_constr_bad, 23, 26, 28
ci_rbod_constr_bad_Q, 25
ci_rbod_constr_Q, 27
ci_rbod_dir, 29
ci_rbod_spatial, 30
ci_smaa_constr, 31
ci_wroclaw, 33
Compind-package, 2

EU_2020, 34
EU_NUTS1, 36

normalise_ci, 6, 21, 22, 36

rank, 37