Package ‘REAT’

October 12, 2022

Type Package
Title Regional Economic Analysis Toolbox
Version 3.0.3
Date 2021-08-24
Author Thomas Wieland
Maintainer Thomas Wieland <thomas.wieland.geo@gmail.com>
Description Collection of models and analysis methods used in regional and urban economics and (quantitative) economic geography, e.g. measures of inequality, regional disparities and convergence, regional specialization as well as accessibility and spatial interaction models.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2021-09-03 08:00:11 UTC

R topics documented:

REAT-package .......................................................... 3
atkinson .............................................................. 4
Automotive .......................................................... 5
betaconv.nls .......................................................... 6
betaconv.ols .......................................................... 9
betaconv.speed ....................................................... 12
cnc ................................................................. 14
converse ............................................................ 15
coulter ............................................................... 16
curvefit ............................................................. 17
cv ................................................................. 19
dalton ............................................................... 21
disp ............................................................... 22
dist.buf ............................................................. 23
dist.calc ........................................................... 25
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>dist.mat</td>
<td>26</td>
</tr>
<tr>
<td>durpug</td>
<td>27</td>
</tr>
<tr>
<td>ellison.a</td>
<td>28</td>
</tr>
<tr>
<td>ellison.a2</td>
<td>29</td>
</tr>
<tr>
<td>ellison.c</td>
<td>30</td>
</tr>
<tr>
<td>ellison.c2</td>
<td>31</td>
</tr>
<tr>
<td>EU28.emp</td>
<td>33</td>
</tr>
<tr>
<td>FK2014_EGC</td>
<td>34</td>
</tr>
<tr>
<td>Freiburg</td>
<td>35</td>
</tr>
<tr>
<td>G.counties.gdp</td>
<td>36</td>
</tr>
<tr>
<td>G.regions.emp</td>
<td>39</td>
</tr>
<tr>
<td>G.regions.industries</td>
<td>40</td>
</tr>
<tr>
<td>gifpro</td>
<td>41</td>
</tr>
<tr>
<td>gifpro.tbs</td>
<td>44</td>
</tr>
<tr>
<td>gini</td>
<td>46</td>
</tr>
<tr>
<td>gini.conc</td>
<td>50</td>
</tr>
<tr>
<td>gini.spec</td>
<td>52</td>
</tr>
<tr>
<td>gini2</td>
<td>55</td>
</tr>
<tr>
<td>Goettingen</td>
<td>57</td>
</tr>
<tr>
<td>GoettingenHealth1</td>
<td>59</td>
</tr>
<tr>
<td>GoettingenHealth2</td>
<td>60</td>
</tr>
<tr>
<td>growth</td>
<td>61</td>
</tr>
<tr>
<td>hansen</td>
<td>62</td>
</tr>
<tr>
<td>herf</td>
<td>65</td>
</tr>
<tr>
<td>hoover</td>
<td>67</td>
</tr>
<tr>
<td>howard.cl</td>
<td>68</td>
</tr>
<tr>
<td>howard.xcl</td>
<td>69</td>
</tr>
<tr>
<td>howard.xcl2</td>
<td>70</td>
</tr>
<tr>
<td>huff</td>
<td>71</td>
</tr>
<tr>
<td>krugman.conc</td>
<td>74</td>
</tr>
<tr>
<td>krugman.conc2</td>
<td>75</td>
</tr>
<tr>
<td>krugman.spec</td>
<td>77</td>
</tr>
<tr>
<td>krugman.spec2</td>
<td>78</td>
</tr>
<tr>
<td>litzenberger</td>
<td>79</td>
</tr>
<tr>
<td>litzenberger2</td>
<td>80</td>
</tr>
<tr>
<td>lm.beta</td>
<td>82</td>
</tr>
<tr>
<td>locq</td>
<td>83</td>
</tr>
<tr>
<td>locq.growth</td>
<td>85</td>
</tr>
<tr>
<td>locq2</td>
<td>87</td>
</tr>
<tr>
<td>lorenz</td>
<td>89</td>
</tr>
<tr>
<td>mean2</td>
<td>92</td>
</tr>
<tr>
<td>mssd</td>
<td>93</td>
</tr>
<tr>
<td>portfolio</td>
<td>94</td>
</tr>
<tr>
<td>rca</td>
<td>96</td>
</tr>
<tr>
<td>reilly</td>
<td>101</td>
</tr>
<tr>
<td>ripley</td>
<td>103</td>
</tr>
<tr>
<td>sd2</td>
<td>105</td>
</tr>
<tr>
<td>shift</td>
<td>107</td>
</tr>
</tbody>
</table>
Description

In regional and urban economics and economic geography, very frequent research fields are the existence and evolution of agglomerations due to (internal and external) agglomeration economies, regional economic growth and regional disparities, where these concepts and relationships are closely related to each other (Capello/Nijkamp 2009, Dinc 2015, Farhauer/Kroell 2013, McCann/van Oort 2009). Also accessibility and spatial interaction modeling is mostly regarded as related to these disciplines (Aoyama et al. 2011, Guessefeldt 1999). The group of the related analysis methods is sometimes summarized by the term regional analysis or regional economic analysis (Dinc 2015, Guessefeldt 1999, Isard 1960).

This package contains a collection of models and analysis methods used in regional and urban economics and (quantitative) economic geography. The functions in this package can be divided into seven groups:

1. Inequality, concentration and dispersion, including Gini coefficient, Lorenz curve, Herfindahl-Hirschman-coefficient, Theil coefficient, Hoover coefficient and (weighted) coefficient of variation
2. Specialization of regions and spatial concentration of industries, including location quotient, spatial Gini coefficients for regional specialization and industry concentration and Krugman coefficients for regional specialization and industry concentration
3. Regional disparities and regional convergence, especially analysis of beta and sigma convergence for cross-sectional data
4. Regional growth, including portfolio matrix, several types of shift-share analysis and commercial area prognosis ("GIFPRO")
5. Spatial interaction and accessibility models, including Huff model and Hansen accessibility
6. Proximity analysis, including calculation of distance matrices and buffers
7. Additional tools for data preparation und visualization, such as for creating dummy variables and calculating standardized regression coefficients. The package also contains data examples.
atkinson

Author(s)

Thomas Wieland
Maintainer: Thomas Wieland <thomas.wieland.geo@googlemail.com>

References


atkinson

Atkinson Inequality Index

Description

Calculating the Atkinson Inequality Index e.g. with respect to regional income

Usage

atkinson(x, epsilon = 0.5, na.rm = TRUE)

Arguments

x A numeric vector (e.g. dataset of regional income)
epsilon A single value of the \( \epsilon \) weighting coefficient (default: at. epsilon = 0.5)
na.rm logical argument that indicates whether NA values should be excluded before computing results

Details

The Atkinson Inequality Index (AI) varies between 0 (no inequality/concentration) and 1 (complete inequality/concentration). It can be used for economic inequality and/or regional disparities (Portnov/Felsenstein 2010).
Automotive

Value
A single numeric value of the Atkinson Inequality Index ($0 < AI < 1$).

Author(s)
Thomas Wieland

References

See Also
cv, gini, gini2, herf, theil, hoover, coulter, dalton, disp

Examples
atkinson(c(100,0,0,0), epsilon = 0.8)
atkinson(c(100,100,100,100), epsilon = 0.8)

Automotive industry data

Description
Top 20 automotive industry companies, including their manufacturing quantity and turnovers (Table from wikipedia)

Usage
data("Automotive")

Format
A data frame with 20 observations on the following 8 variables.

Rank  Rank of the company
Company  Name of the company (German)
Country  Origin county of the company (German)
Quantity2014  Quantity of produced vehicles in 2014
Quantity2014_car  Quantity of produced cars in 2014
Turnover2008  Annual turnover 2008 (in billion dollars)
Turnover2012  Annual turnover 2012 (in billion dollars)
Turnover2013  Annual turnover 2013 (in billion dollars)
Source


References


Examples

# Market concentration in automotive industry

data(Automotive)

gini(Automotive$Turnover2008, lsize=1, lc=TRUE, le.col = “black”, 
lc.col = “orange”, lcx = “Shares of companies”, lcy = “Shares of turnover / cars”, 
lctitle = “Automotive industry: market concentration”, 
lcg = TRUE, lcgn = TRUE, lc.caption = “Turnover 2008:”, lcg.lab.x = 0, lcg.lab.y = 1)

# Gini coefficient and Lorenz curve for turnover 2008

gini(Automotive$Turnover2013, lsize=1, lc = TRUE, add.lc = TRUE, lc.col = “red”, 
lcg = TRUE, lcgn = TRUE, lc.caption = “Turnover 2013:”, lcg.lab.x = 0, lcg.lab.y = 0.85)

# Adding Gini coefficient and Lorenz curve for turnover 2013

gini(Automotive$Quantity2014_car, lsize=1, lc = TRUE, add.lc = TRUE, lc.col = “blue”, 
lcg = TRUE, lcgn = TRUE, lc.caption = “Cars 2014:”, lcg.lab.x = 0, lcg.lab.y = 0.7)

# Adding Gini coefficient and Lorenz curve for cars 2014

---

betaconv.nls Analysis of regional beta convergence using nonlinear regression

Description

This function provides the analysis of absolute and conditional regional economic beta convergence for cross-sectional data using a nonlinear least squares (NLS) technique.

Usage

betaconv.nls(gdp1, time1, gdp2, time2, conditions = NULL, conditions.formula = NULL, 
conditions.startval = NULL, beta.plot = FALSE, beta.plotPSize = 1, 
beta.plotPCol = ”black”, beta.plotLine = FALSE, beta.plotLineCol = ”red”, 
beta.plotX = ”Ln (initial)”, beta.plotY = ”Ln (growth)”,
beta.plotTitle = ”Beta convergence”, beta.bgCol = ”gray95”, beta.bgrid = TRUE, 
beta.bgridCol = ”white”, beta.bgridSize = 2, beta.bgridType = ”solid”,
print.results = TRUE)
Arguments

- **gdp1**: A numeric vector containing the GDP per capita (or another economic variable) at time $t$.
- **time1**: A single value of time $t$ (= the initial year).
- **gdp2**: A numeric vector containing the GDP per capita (or another economic variable) at time $t+1$ or a data frame containing the GDPs per capita (or another economic variable) at time $t+1, t+2, t+3, \ldots, t+n$.
- **time2**: A single value of time $t+1$ or $t+n$, respectively.
- **conditions**: A data frame containing the conditions for conditional beta convergence.
- **conditions.formula**: A formula for the functional linkage of the conditions in the case of conditional beta convergence.
- **conditions.startval**: Starting values for the parameters of the conditions in the case of conditional beta convergence.
- **beta.plot**: Boolean argument that indicates if a plot of beta convergence has to be created.
- **beta.plotPSize**: If beta.plot = TRUE: Point size in the beta convergence plot.
- **beta.plotPCol**: If beta.plot = TRUE: Point color in the beta convergence plot.
- **beta.plotLine**: If beta.plot = TRUE: Logical argument that indicates if a regression line has to be added to the plot.
- **beta.plotLineCol**: If beta.plot = TRUE and beta.plotLine = TRUE: Line color of regression line.
- **beta.plotX**: If beta.plot = TRUE: Name of the X axis.
- **beta.plotY**: If beta.plot = TRUE: Name of the Y axis.
- **beta.plotTitle**: If beta.plot = TRUE: Plot title.
- **beta.bgCol**: If beta.plot = TRUE: Plot background color.
- **beta.bgrid**: If beta.plot = TRUE: Logical argument that indicates if the plot contains a grid.
- **beta.bgridCol**: If beta.plot = TRUE and beta.bgrid = TRUE: Color of the grid.
- **beta.bgridSize**: If beta.plot = TRUE and beta.bgrid = TRUE: Size of the grid.
- **beta.bgridType**: If beta.plot = TRUE and beta.bgrid = TRUE: Type of the grid.
- **print.results**: Logical argument that indicates if the function shows the results or not.

Details

From the regional economic perspective (in particular the neoclassical growth theory), regional disparities are expected to decline. This convergence can have different meanings: Sigma convergence ($\sigma$) means a harmonization of regional economic output or income over time, while beta convergence ($\beta$) means a decline of dispersion because poor regions have a stronger economic growth than rich regions (Capello/Nijkamp 2009). Regardless of the theoretical assumptions of a harmonization in reality, the related analytical framework allows to analyze both types of convergence for cross-sectional data (GDP p.c. or another economic variable, $y$, for $i$ regions and two points in time, $t$ and $t + T$), or one starting point ($t$) and the average growth within the following $n$ years.
(t + 1, t + 2, ..., t + n), respectively. Beta convergence can be calculated either in a linearized OLS regression model or in a nonlinear regression model. When no other variables are integrated in this model, it is called absolute beta convergence. Implementing other region-related variables (conditions) into the model leads to conditional beta convergence. If there is beta convergence ($\beta < 0$), it is possible to calculate the speed of convergence, $\lambda$, and the so-called Half-Life $H$, while the latter is the time taken to reduce the disparities by one half (Allington/McCombie 2007, Goecke/Huether 2016). There is sigma convergence, when the dispersion of the variable ($\sigma$), e.g. calculated as standard deviation or coefficient of variation, reduces from $t$ to $t + T$. This can be measured using ANOVA for two years or trend regression with respect to several years (Furceri 2005, Goecke/Huether 2016).

This function calculates absolute and/or conditional beta convergence using a nonlinear least squares approach for estimation. It needs at least two vectors (GDP p.c. or another economic variable, $y$, for $i$ regions) and the related two points in time ($t$ and $t + T$). If the beta coefficient is negative (using OLS) or positive (using NLS), there is beta convergence.

Value

A list containing the following objects:

- **regdata** A data frame containing the regression data, including the ln-transformed economic variables
- **abeta** A list containing the estimates of the absolute beta convergence regression model, including lambda and half-life
- **cbeta** If conditions are stated: a list containing the estimates of the conditional beta convergence regression model, including lambda and half-life

Author(s)

Thomas Wieland

References


See Also

rca, betaconv.ols, betaconv.speed, sigmaconv, sigmaconv.t, cv, sd2, var2

Examples

data (G.counties.gdp)
# Loading GDP data for Germany (counties = Landkreise)
conditions = NULL, print.results = TRUE)
# Two years, no conditions (Absolute beta convergence)

---

betaconv.ols  
Analysis of regional beta convergence using OLS regression

Description

This function provides the analysis of absolute and conditional regional economic beta convergence
for cross-sectional data using ordinary least squares (OLS) technique.

Usage

betaconv.ols(gdp1, time1, gdp2, time2, conditions = NULL, beta.plot = FALSE,
beta.plotPSize = 1, beta.plotPCol = "black", beta.plotLine = FALSE,
beta.plotLineCol = "red", beta.plotX = "Ln (initial)", beta.plotY = "Ln (growth)",
beta.plotTitle = "Beta convergence", beta.bgCol = "gray95", beta.bgrid = TRUE,
beta.bgridCol = "white", beta.bgridSize = 2, beta.bgridType = "solid",
print.results = FALSE)

Arguments

gdp1  A numeric vector containing the GDP per capita (or another economic variable)
at time t

time1  A single value of time t (= the initial year)

gdp2  A numeric vector containing the GDP per capita (or another economic variable)
at time t+1 or a data frame containing the GDPs per capita (or another economic
variable) at time t+1, t+2, t+3, ..., t+n
time2  A single value of time t+1 or t+n, respectively

conditions  A data frame containing the conditions for conditional beta convergence

beta.plot  Boolean argument that indicates if a plot of beta convergence has to be created

beta.plotPSize  If beta.plot = TRUE: Point size in the beta convergence plot

beta.plotPCol  If beta.plot = TRUE: Point color in the beta convergence plot

beta.plotLine  If beta.plot = TRUE: Logical argument that indicates if a regression line has to
be added to the plot

beta.plotLineCol  If beta.plot = TRUE and beta.plotLine = TRUE: Line color of regression line
beta.plotX  If beta.plot = TRUE: Name of the X axis
beta.plotY  If beta.plot = TRUE: Name of the Y axis
beta.plotTitle  If beta.plot = TRUE: Plot title
beta.bgCol  If beta.plot = TRUE: Plot background color
beta.bgrid  If beta.plot = TRUE: Logical argument that indicates if the plot contains a grid
beta.bgridCol  If beta.plot = TRUE and beta.bgrid = TRUE: Color of the grid
beta.bgridSize  If beta.plot = TRUE and beta.bgrid = TRUE: Size of the grid
beta.bgridType  If beta.plot = TRUE and beta.bgrid = TRUE: Type of the grid
print.results  Logical argument that indicates if the function shows the results or not

Details

From the regional economic perspective (in particular the neoclassical growth theory), regional disparities are expected to decline. This convergence can have different meanings: Sigma convergence ($\sigma$) means a harmonization of regional economic output or income over time, while beta convergence ($\beta$) means a decline of dispersion because poor regions have a stronger economic growth than rich regions (Capello/Nijkamp 2009). Regardless of the theoretical assumptions of a harmonization in reality, the related analytical framework allows to analyze both types of convergence for cross-sectional data (GDP p.c. or another economic variable, $y$, for $i$ regions and two points in time, $t$ and $t + T$), or one starting point ($t$) and the average growth within the following $n$ years ($t + 1, t + 2, ..., t + n$), respectively. Beta convergence can be calculated either in a linearized OLS regression model or in a nonlinear regression model. When no other variables are integrated in this model, it is called absolute beta convergence. Implementing other region-related variables (conditions) into the model leads to conditional beta convergence. If there is beta convergence ($\beta < 0$), it is possible to calculate the speed of convergence, $\lambda$, and the so-called Half-Life $H$, while the latter is the time taken to reduce the disparities by one half (Allington/McCombie 2007, Goecke/Huether 2016). There is sigma convergence, when the dispersion of the variable ($\sigma$), e.g. calculated as standard deviation or coefficient of variation, reduces from $t$ to $t + T$. This can be measured using ANOVA for two years or trend regression with respect to several years (Furceri 2005, Goecke/Huether 2016).

This function calculates absolute and/or conditional beta convergence using ordinary least squares regression (OLS) for estimation. It needs at least two vectors (GDP p.c. or another economic variable, $y$, for $i$ regions) and the related two points in time ($t$ and $t + T$). If the beta coefficient is negative (using OLS) or positive (using NLS), there is beta convergence.

Value

A list containing the following objects:

regdata  A data frame containing the regression data, including the ln-transformed economic variables
abeta  A list containing the estimates of the absolute beta convergence regression model, including lambda and half-life
cbeta  If conditions are stated: a list containing the estimates of the conditional beta convergence regression model, including lambda and half-life
Author(s)

Thomas Wieland

References


See Also

rca, betaconv.nls, betaconv.speed, sigmaconv, sigmaconv.t, cv, sd2, var2

Examples

data (G.counties.gdp)

# Two years, no conditions (Absolute beta convergence)

regionaldummies <- to.dummy(G.counties.gdp$regional)
# Creating dummy variables for West/East
G.counties.gdp$West <- regionaldummies[,2]
G.counties.gdp$East <- regionaldummies[,1]
# Adding dummy variables to data

# Two years, with condition (dummy for West/East)
# (Absolute and conditional beta convergence)

# Store results in object
betaconverg1$cbeta$estimates
# Addressing estimates for the conditional beta model


betaconv.ols (G.counties.gdp$gdppc2010, 2010, G.counties.gdp[65:66], 2012, conditions = G.counties.gdp[c(70,71)], print.results = TRUE)  # Three years (2010–2012), with conditions (Absolute and conditional beta convergence)

betaconverg2 <- betaconv.ols (G.counties.gdp$gdppc2010, 2010, G.counties.gdp[65:66], 2012, conditions = G.counties.gdp[c(70,71)], print.results = TRUE)  # Store results in object

betaconverg2$cbeta$estimates

---

**betaconv.speed**

*Regional beta convergence: Convergence speed and half-life*

### Description

This function calculates the beta convergence speed and half-life based on a given beta value and time interval.

### Usage

```
betaconv.speed(beta, tinterval, print.results = TRUE)
```

### Arguments

- **beta**: Beta value
- **tinterval**: Time interval (in time units, such as years)
- **print.results**: Logical argument that indicates if the function shows the results or not

### Details

From the regional economic perspective (in particular the neoclassical growth theory), regional disparities are expected to decline. This *convergence* can have different meanings: *Sigma convergence* ($\sigma$) means a harmonization of regional economic output or income over time, while *beta convergence* ($\beta$) means a decline of dispersion because poor regions have a stronger economic growth than rich regions (Capello/Nijkamp 2009). Regardless of the theoretical assumptions of a harmonization in reality, the related analytical framework allows to analyze both types of convergence for cross-sectional data (GDP p.c. or another economic variable, $y$, for $i$ regions and two points in time, $t$ and $t + T$), or one starting point ($t$) and the average growth within the following $n$ years ($t + 1, t + 2, \ldots, t + n$), respectively. Beta convergence can be calculated either in a linearized OLS regression model or in a nonlinear regression model. When no other variables are integrated in this model, it is called *absolute* beta convergence. Implementing other region-related variables
(conditions) into the model leads to conditional beta convergence. If there is beta convergence ($\beta < 0$), it is possible to calculate the speed of convergence, $\lambda$, and the so-called Half-Life $H$, while the latter is the time taken to reduce the disparities by one half (Allington/McCombie 2007, Goecke/Huether 2016). There is sigma convergence, when the dispersion of the variable ($\sigma$), e.g. calculated as standard deviation or coefficient of variation, reduces from $t$ to $t + T$. This can be measured using ANOVA for two years or trend regression with respect to several years (Furceri 2005, Goecke/Huether 2016).

This function calculates the speed of convergence, $\lambda$, and the Half-Life, $H$, based on a given $\beta$ value and time interval.

Value

A matrix containing the following objects:

- Lambda: Lambda value (convergence speed)
- Half-Life: Half-life values

Author(s)

Thomas Wieland

References


See Also

betaconv.nls, betaconv.ols, sigmaconv, sigmaconv.t, cv, sd2, var2
Examples

```r
speed <- betaconv.speed(-0.008070533, 1)
speed[1] # lambda
speed[2] # half-life
```

### conc

**Measures of industry concentration**

### Description

Calculating three measures of industry concentration (Gini, Krugman, Hoover) for a set of \( I \) industries

### Usage

```r
conc(e_ij, industry.id, region.id, na.rm = TRUE)
```

### Arguments

- `e_ij`: a numeric vector with the employment of the industry \( i \) in region \( j \)
- `industry.id`: a vector containing the IDs of the industries \( i \)
- `region.id`: a vector containing the IDs of the regions \( j \)
- `na.rm`: logical argument that indicates whether NA values should be excluded before computing results

### Details

This function is a convenient wrapper for all functions calculating measures of spatial concentration of industries (Gini, Krugman, Hoover)

### Value

A matrix with three columns (Gini coefficient, Krugman coefficient, Hoover coefficient) and \( I \) rows (one for each regarded industry).

### Author(s)

Thomas Wieland

### References

converse

See Also

gini.conc, krugman.conc2, hoover

Examples

data(G.regions.industries)

conc_i <- conc(e_ij = G.regions.industries$emp_all,
industry.id = G.regions.industries$ind_code,
region.id = G.regions.industries$region_code)

converse

Breaking point formula by Converse

Description

Calculating the breaking point between two cities or retail locations

Usage

converse(P_a, P_b, D_ab)

Arguments

P_a    a single numeric value of attractivity/population size of location/city a
P_b    a single numeric value of attractivity/population size of location/city b
D_ab   a single numeric value of the transport costs (e.g. distance) between a and b

Details

The breaking point formula by Converse (1949) is a modification of the law of retail gravitation by Reilly (1929, 1931) (see the functions reilly and reilly.lambda). The aim of the calculation is to determine the boundaries of the market areas between two locations/cities in consideration of their attractivity/population size and the transport costs (e.g. distance) between them. The models by Reilly and Converse are simple spatial interaction models and are considered as deterministic market area models due to their exact allocation of demand origins to locations. A probabilistic approach including a theoretical framework was developed by Huff (1962) (see the function huff).

Value

a list with two values (B_a: distance from location a to breaking point, B_b: distance from location b to breaking point)

Author(s)

Thomas Wieland
References


See Also

huff, reilly

Examples

# Example from Huff (1962):
converse (400000, 200000, 80)  
# two cities (population 400.000 and 200.000 with a distance separating them of 80 miles)

coulters

Coulter Coefficient

Description

Calculating the Coulter Coefficient e.g. with respect to regional income

Usage

coulters(x, weighting = NULL, na.rm = TRUE)

Arguments

x A numeric vector (e.g. dataset of regional income)
weighting a weighting vector, e.g. population
na.rm logical argument that indicates whether NA values should be excluded before computing results
Details

The Coulter Coefficient (CC) varies between 0 (no inequality/concentration) and 1 (complete inequality/concentration). It can be used for economic inequality and/or regional disparities (Portnov/Felsenstein 2010).

Value

A single numeric value of the Coulter Coefficient (0 < CC < 1).

Author(s)

Thomas Wieland

References


See Also

cv, gini, gini2, herf, theil, hoover, atkinson, dalton, disp

Examples

bip <- c(400, 400, 400, 400, NA)
bev <- c(1, 1, 1, 200, NA)
coulter(bip, bev)
Arguments

- **x**: a numeric vector containing the explanatory variable
- **y**: a numeric vector containing the dependent variable
- **y.max**: Optional: given maximum for the logistic regression function
- **extrapol**: a single numeric value for how many x units the dependent variable y shall be extrapolated
- **plot.curves**: Logical argument that indicates whether the curves shall be plotted or not
- **pcol**: If plot.curves = TRUE: Point color
- **ptype**: If plot.curves = TRUE: Point type (pch)
- **psize**: If plot.curves = TRUE: Point size
- **lin.col**: If plot.curves = TRUE: Color of linear regression line
- **pow.col**: If plot.curves = TRUE: Color of power function regression line
- **exp.col**: If plot.curves = TRUE: Color of exponential function regression line
- **logi.col**: If plot.curves = TRUE: Color of logistic function regression line
- **plot.title**: If plot.curves = TRUE: Plot title
- **plot.legend**: If plot.curves = TRUE: Logical argument that indicates whether a legend is added to the plot or not
- **xlab**: If plot.curves = TRUE: X axis label
- **ylab**: If plot.curves = TRUE: Y axis label
- **y.min**: Optional: Y axis minimum
- **...**: Optional: other plot parameters
- **print.results**: Logical argument that indicates whether the model results are shown or not

Details

Curve fitting for a given independent and dependent variable \(y = f(x)\). Similar to curve fitting in SPSS or Excel. Fitting of nonlinear regression models (power, exponential, logistic) via intrinsically linear models (Rawlings et al. 1998).

Value

A data frame containing the regression results (Parameters a and b, std. errors, t values, ...)

Author(s)

Thomas Wieland

References

Examples

```r
x <- 1:20
y <- 3-2*x
curvefit(x, y, plot.curves = TRUE)
  # fit with plot
curvefit(x, y, extrapol=10, plot.curves = TRUE)
  # fit and extrapolation with plot

x <- runif(20, min = 0, max = 100)
  # some random data

# linear
y_resid <- runif(20, min = 0, max = 10)
  # random residuals
y <- 3+(-0.112*x)+y_resid
curvefit(x, y)

# power
y_resid <- runif(20, min = 0.1, max = 0.2)
  # random residuals
y <- 3*(x^-0.112)*y_resid
curvefit(x, y)

# exponential
y_resid <- runif(20, min = 0.1, max = 0.2)
  # random residuals
y <- 3*exp(-0.112*x)*y_resid
curvefit(x, y)

# logistic
y_resid <- runif(20, min = 0.1, max = 0.2)
  # random residuals
y <- 100/(1+exp(3*(-0.112*x)))*y_resid
curvefit(x, y)
```

---

### cv

**Coefficient of variation**

**Description**

Calculating the coefficient of variation (cv), standardized and non-standardized, weighted and non-weighted

**Usage**

```r
cv (x, is.sample = TRUE, coefnorm = FALSE, weighting = NULL, wmean = FALSE, na.rm = TRUE)
```
Arguments

- **x**: a numeric vector
- **is.sample**: logical argument that indicates if the dataset is a sample or the population (default: **is.sample** = TRUE, so the denominator of variance is \(n - 1\))
- **coefnorm**: logical argument that indicates if the function output is the standardized cv (\(0 < v^* < 1\)) or not (\(0 < v < \infty\)) (default: **coefnorm** = FALSE)
- **weighting**: a numeric vector containing weighting data to compute the weighted coefficient of variation (instead of the non-weighted cv)
- **wmean**: logical argument that indicates if the weighted mean is used when calculating the weighted coefficient of variation
- **na.rm**: logical argument that whether NA values should be extracted or not

Details

The coefficient of variation, \(v\), is a dimensionless measure of statistical dispersion (\(0 < v < \infty\)), based on variance and standard deviation, respectively. From a regional economic perspective, it is closely linked to the concept of sigma convergence (\(\sigma\)) which means a harmonization of regional economic output or income over time, while the other type of convergence, beta convergence (\(\beta\)), means a decline of dispersion because poor regions have a stronger growth than rich regions (Capello/Nijkamp 2009). The cv allows to summarize regional disparities (e.g. disparities in regional GDP per capita) in one indicator and is more frequently used for this purpose than the standard deviation, especially in analyzing of \(\sigma\) convergence over a long period (e.g. Lessmann 2005, Huang/Leung 2009, Siljak 2015). But the cv can also be used for any other types of disparities or dispersion, such as disparities in supply (e.g. density of physicians or grocery stores).

The cv (variance, standard deviation) can be weighted by using a second weighting vector. As there is more than one way to weight measures of statistical dispersion, this function uses the formula for the weighted cv (\(v_w\)) from Sheret (1984). The cv can be standardized, while this function uses the formula for the standardized cv (\(v^*\), with \(0 < v^* < 1\)) from Kohn/Oeztuerk (2013). The vector \(x\) is automatically treated as a sample (such as in the base \(sd\) function), so the denominator of variance is \(n - 1\), if it is not, set **is.sample** = FALSE.

Value

Single numeric value. If **coefnorm** = FALSE the function returns the non-standardized cv (\(0 < v < \infty\)). If **coefnorm** = TRUE the standardized cv (\(0 < v^* < 1\)) is returned.

Author(s)

Thomas Wieland

References


Huang, Y./Leung, Y. (2009): “Measuring Regional Inequality: A Comparison of Coefficient of Variation and Hoover Concentration Index”. In: The Open Geography Journal, 2, p. 25-34.


See Also
gini, herf, hoover, rca

Examples

# Regional disparities / sigma convergence in Germany
data(G.counties.gdp)
# GDP per capita for German counties (Landkreise)
cvs <- apply(G.counties.gdp[54:68], 2, FUN = cv)
# Calculating cv for the years 2000-2014
years <- 2000:2014
plot(years, cvs, "l", ylim=c(0.3,0.6), xlab = "year", ylab = "CV of GDP per capita")
# Plot cv over time

---

dalton

**Dalton Inequality Index**

**Description**

Calculating the Dalton Inequality Index e.g. with respect to regional income

**Usage**

dalton(x, na.rm = TRUE)

**Arguments**

- **x**: A numeric vector (e.g. dataset of regional income)
- **na.rm**: logical argument that indicates whether NA values should be excluded before computing results
Details

The Dalton Inequality Index ($\delta$) can be used for economic inequality and/or regional disparities (Portnov/Felsenstein 2010).

Value

A single numeric value of the Dalton Inequality Index.

Author(s)

Thomas Wieland

References


See Also

cv, gini, gini2, herf, theil, hoover, coulter, dalton, disp

Examples

dalton (c(10,10,10,10))
dalton (c(10,0,0,0))
dalton (c(10,1,1,1))

disp

Concentration/inequality/dispersion measures

Description

Calculating a set of concentration/inequality/dispersion measures

Usage

disp(x, weighting = NULL, at.epsilon = 0.5, na.rm = TRUE)

Arguments

x a numeric vector or matrix or columns from a data frame
weighting a weighting vector, e.g. population
at.epsilon Weighting parameter $\epsilon$ for the Atkinson index
na.rm logical argument that indicates whether NA values should be excluded before computing results
Details

This function is a convenient wrapper for all functions calculating concentration/inequality measures.

Value

A matrix containing the concentration/inequality measures.

Author(s)

Thomas Wieland

References


See Also

atkinson, coulter, dalton, cv, gini2, herf, hoover, sd2, theil, williamson

Examples

data(Automotive)

disp(Automotive$Turnover2008)
disp(Automotive[4:8])

dist.buf

Counting points in a buffer

Description

Counting points within a buffer of a given distance with points with given coordinates

Usage

dist.buf(startpoints, sp_id, lat_start, lon_start, endpoints, ep_id, lat_end, lon_end, ep_sum = NULL, bufdist = 500, extract_local = TRUE, unit = "m")
Arguments

startpoints  A data frame containing the start points
sp_id       Column containing the IDs of the startpoints in the data frame startpoints
lat_start   Column containing the latitudes of the start points in the data frame startpoints
lon_start   Column containing the longitudes of the start points in the data frame startpoints
endpoints   A data frame containing the points to count
ep_id       Column containing the IDs of the points to count in the data frame endpoints
lat_end     Column containing the latitudes of the points to count in the data frame endpoints
lon_end     Column containing the longitudes of the points to count in the data frame endpoints
ep_sum      Column of an additional variable in the data frame endpoints to sum
bufdist     The buffer distance
extract_local Logical argument that indicates if the start points should be included or not (default: TRUE)
unit        Unit of the buffer distance: unit="m" for meters, unit="km" for kilometers or unit="miles" for miles

Details

The function is based on the idea of a buffer analysis in GIS (Geographic Information System), e.g. to count the points of interest within a given buffer distance.

Value

The function returns a list containing:

count_table  A data.frame containing two columns: The start point IDs (from) and the number of counted points in the given buffer distance (count_location)
distmat      A data.frame containing the corresponding distance matrix wiht IxJ rows

Author(s)

Thomas Wieland

References


Krider, R. E./Putler, R. S. (2013): “Which Birds of a Feather Flock Together? Clustering and Avoidance Patterns of Similar Retail Outlets”. In: Geographical Analysis, 45, 2, p. 123-149

See Also

dist, dist.mat
**dist.calc**

Euclidean distance between coordinates

**Description**

Calculation of the euclidean distance between two points with stated coordinates (lat, lon)

**Usage**

```r
dist.calc(lat1, lon1, lat2, lon2, unit = "km")
```

**Arguments**

- `lat1`: Latitude of the regarded start point
- `lon1`: Longitude of the regarded start point
- `lat2`: Latitude of the regarded end point
- `lon2`: Longitude of the regarded end point
- `unit`: Unit of the resulting distance: `unit="m"` for meters, `unit="km"` for kilometers or `unit="miles"` for miles

**Value**

A single numeric value

**Author(s)**

Thomas Wieland

**See Also**

`dist.buf, dist.mat`

**Examples**

```r
dist.calc(51.556307, 9.947375, 49.009603, 8.417004) # about 304 kilometers
```
**Description**

Calculation of an euclidean distance matrix between points with stated coordinates (lat, lon)

**Usage**

```r
dist.mat(startpoints, sp_id, lat_start, lon_start, endpoints, ep_id, lat_end, lon_end, unit = "km")
```

**Arguments**

- `startpoints`: A data frame containing the start points
- `sp_id`: Column containing the IDs of the startpoints in the data frame `startpoints`
- `lat_start`: Column containing the latitudes of the start points in the data frame `startpoints`
- `lon_start`: Column containing the longitudes of the start points in the data frame `startpoints`
- `endpoints`: A data frame containing the end points
- `ep_id`: Column containing the IDs of the endpoints in the data frame `endpoints`
- `lat_end`: Column containing the latitudes of the end points in the data frame `endpoints`
- `lon_end`: Column containing the longitudes of the end points in the data frame `endpoints`
- `unit`: Unit of the resulting distance: `unit="m"` for meters, `unit="km"` for kilometers or `unit="miles"` for miles

**Details**

The function calculates an euclidean distance matrix between points with stated coordinates (lat and lon). While \( m \) start points and \( n \) end points are given, the output is a linear \( m \times n \) distance matrix.

**Value**

The function returns a `data.frame` containing 4 columns: The start point IDs (`from`), the end point IDs (`to`), the combination of both (`from_to`) and the calculated distance (`distance`).

**Author(s)**

Thomas Wieland

**References**


Krider, R. E./Putler, R. S. (2013): “Which Birds of a Feather Flock Together? Clustering and Avoidance Patterns of Similar Retail Outlets”. In: Geographical Analysis, 45, 2, p. 123-149
durpug

See Also
dist.dist.buf

Examples

citynames <- c("Goettingen", "Karlsruhe", "Freiburg")
lat <- c(51.556307, 49.009603, 47.9874)
lon <- c(9.947375, 8.417004, 7.89455)
citynames <- c("Goettingen", "Karlsruhe", "Freiburg")
cities <- data.frame(citynames, lat, lon)
dist.mat(cities, "citynames", "lat", "lon")
# Euclidean distance matrix (3 x 3 cities = 9 distances)
dist.buf(cities, "citynames", "lat", "lon", bufdist = 300000)
# Cities within 300 km

---

**durpug**  
*Relative diversity index by Duranton and Puga*

**Description**

Calculating the relative diversity index (RDI) by Duranton and Puga based on regional industry data (normally employment data)

**Usage**

durpug(e_ij, e_i)

**Arguments**

e_ij  
a numeric vector with the employment of the industries $i$ in region $j$
e_i  
a numeric vector with the all-over employment in the industries $i$

**Value**

A single numeric value of $RDI$

**Author(s)**

Thomas Wieland

**References**


See Also
gini.spec, krugman.spec, hoover

Examples

# Example Goettingen:
data(Goettingen)
# Loads the data
durpug (Goettingen$Goettingen2008[2:13], Goettingen$BRD2008[2:13])
# Returns the Duranton-Puga RDI for Goettingen

def ellison.a

Ellison-Glaeser Agglomeration Index

Description

Calculating the Agglomeration Index by Ellison and Glaeser for a single industry \( i \).

Usage

ellison.a(e_ik, e_j, regions, print.results = TRUE)

Arguments

e_ik a numeric vector containing the no. of employees of firm \( k \) from industry \( i \)
e_j a numeric vector containing the no. of employees in the regions \( j \)
regions a vector containing the IDs/names of the regions \( j \)
print.results logical argument that indicates whether the function prints the results or not (only for internal use)

Details

The Ellison-Glaeser Agglomeration Index is not standardized. A value of \( \gamma_i = 0 \) indicates a spatial distribution of firms equal to a dartboard approach. Values below zero indicate spatial dispersion, values greater than zero indicate clustering.

Value

A matrix with five columns \( (\gamma_i, G_i, z_{G_i}, K_i, \text{and } HHI_i) \).

Author(s)

Thomas Wieland
ellison.a2


See Also

  gini.conc, gini.spec, locq, locq2, howard.cl, howard.xcl, howard.xcl2, litzenberger, litzenberger2

Examples

  # Example from Farhauer/Kroell (2014):
  E_ik <- c(200,650,12000,100,50,16000,13000,1500,1500,25000)
  E_j <- c(500000,400000,100000)
  ellison.a(E_ik, E_j, j)
  # 0.05990628

Description

Calculating the Agglomeration Index by Ellison and Glaeser for a given number of $I$ industries

Usage

  ellison.a2(e_ik, industry, region, print.results = TRUE)

Arguments

  e_ik a numeric vector containing the no. of employees of firm $k$ from industry $i$
  industry a vector containing the IDs/names of the industries $i$
  region a vector containing the IDs/names of the regions $j$
  print.results logical argument that indicates whether the function prints the results or not (only for internal use)

Details

The Ellison-Glaeser Agglomeration Index is not standardized. A value of $\gamma_i = 0$ indicates a spatial distribution of firms equal to a dartboard approach. Values below zero indicate spatial dispersion, values greater than zero indicate clustering.
Value

A matrix with five columns ($\gamma_i$, $G_i$, $z_{Gi}$, $K_i$ and $HHI_i$) and $I$ rows (one for each industry).

Author(s)

Thomas Wieland

References


See Also

ellison.a2, gini.conc, gini.spec, locq, locq2, howard.cl, howard.xcl, howard.xcl2, litzenberger, litzenberger2

Examples

# Example data from Farhauer/Kroell (2014):
data(FK2014_EGC)
ellison.a2(FK2014_EGC$emp_firm, FK2014_EGC$industry, FK2014_EGC$region)

data(FK2014_EGC$emp_firm, FK2014_EGC$industry, FK2014_EGC$region)

ellison.c

Ellison-Glaeser Coagglomeration Index

Description

Calculating the Coagglomeration Index by Ellison and Glaeser for one set of $U$ industries

Usage

ellison.c(e_ik, industry, region, e_j = NULL, c.industries = NULL)

Arguments

e_ik a numeric vector containing the no. of employees of firm $k$ from industry $i$
industry a vector containing the IDs/names of the industries $i$
region a vector containing the IDs/names of the regions $j$
e_j a numeric vector containing the total employment of the regions $j$
c.industries optional: a vector containing the regarded $U$ industries (where $U \leq I$)
### Details

The Ellison-Glaeser Coagglomeration Index is not standardized. A value of $\gamma_c = 0$ indicates a spatial distribution of firms equal to a dartboard approach. Values below zero indicate spatial dispersion, values greater than zero indicate clustering.

### Value

A single value of $\gamma_c$

### Author(s)

Thomas Wieland

### References


### See Also

ellison.a, ellison.a2, ellison.c2, gini.conc, gini.spec, locq, locq2, howard.cl, howard.xcl, howard.xcl2, litzenberger, litzenberger2

### Examples

# Example from Farhauer/Kroell (2014):

data(FK2014_EGC)

ellison.c2(FK2014_EGC$emp_firm, FK2014_EGC$industry, FK2014_EGC$region, FK2014_EGC$emp_region)
Arguments

e_ik          a numeric vector containing the no. of employees of firm \( k \) from industry \( i \)
industry      a vector containing the IDs/names of the industries \( i \)
region        a vector containing the IDs/names of the regions \( j \)
e_j           a numeric vector containing the total employment of the regions \( j \)
print.results logical argument that indicates whether the results are printed or not (for internal use)

Details

The Ellison-Glaeser Coagglomeration Index is not standardized. A value of \( \gamma^c = 0 \) indicates a spatial distribution of firms equal to a dartboard approach. Values below zero indicate spatial dispersion, values greater than zero indicate clustering.

Value

A single value of \( \gamma^c \)

Author(s)

Thomas Wieland

References


See Also

e ellison.a, ellison.a2, ellison.c, gini.conc, gini.spec, locq, locq2, howard.c1, howard.xcl, howard.xcl2, litzenberger, litzenberger2

Examples

# Example from Farhauer/Kroell (2014):
data(FK2014_EGC)

ellison.c2(FK2014_EGC$emp_firm, FK2014_EGC$industry, FK2014_EGC$region, FK2014_EGC$emp_region)
# this may take a while
EU28.emp

Eurostat national employment data 2004-2016

Description

Employment data for EU countries 2004-2016 (Source: Eurostat)

Usage

data("EU28.emp")

Format

A data frame with 3000 observations on the following 7 variables.

- unit: measuring unit: thousand persons (THS_PER)
- nace_r2: NACE industry classification
- s_adj: Adjustment of data: Not seasonally adjusted data (NSA)
- na_item: a factor with levels SAL_DC
- geo: NUTS nation code
- time: year
- emp1000: Industry-specific employment in thousand persons

Source


Examples

data(EU28.emp)
EU28.emp[EU28.emp$time == 2016,]
# only data for 2016
FK2014_EGC  Fictional sample data of 42 firms

Description

Dataset with 42 firms from 4 industries in 3 regions (fictional sample data from Farhauer/Kroell 2014)

Usage

```r
data("FK2014_EGC")
```

Format

A data frame with 42 observations on the following 5 variables.

- `region` unique ID of the region
- `industry` name of the industry (German language)
- `firm` firm ID
- `emp_firm` each firm’s no. of employees
- `emp_region` total employment of the region

Source


References


Examples

```r
# Example from Farhauer/Kroell (2014):
data(FK2014_EGC)
ellison.c(FK2014_EGC$emp_firm, FK2014_EGC$industry,FK2014_EGC$region, FK2014_EGC$emp_region)
```
Data set with industry-specific employment in Freiburg and Germany in the years 2008 and 2014

Usage

```r
data("Freiburg")
```

Format

A data frame with 9 observations on the following 8 variables.

- `industry` a factor with levels for the regarded industry based on the German official economic statistics (WZ2008)
- `e_Freiburg2008` a numeric vector with industry-specific employment in Freiburg 2008
- `e_Freiburg2014` a numeric vector with industry-specific employment in Freiburg 2014
- `e_g_Freiburg_0814` a numeric vector containing the growth of industry-specific employment in Freiburg 2008-2014, percentage
- `e_Germany2008` a numeric vector with industry-specific employment in Germany 2008
- `e_Germany2014` a numeric vector with industry-specific employment in Germany 2014
- `e_g_Germany_0814` a numeric vector containing the growth of industry-specific employment in Germany 2008-2014, percentage
- `color` a factor containing colors (blue, brown, ...)

Source

Statistische Aemter des Bundes und der Laender: Regionaldatenbank Deutschland, Tab. 254-74-4, own calculations

Examples

```r
data(Freiburg)
# Loads the data
growth(Freiburg$e_Freiburg2008, Freiburg$e_Freiburg2014, growth.type = "rate")
# Industry-specific growth rates for Freiburg 2008 to 2014
```
The dataset contains the Gross Domestic Product (GDP) absolute and per capita (in EUR, at current prices) for the 402 German counties (Landkreise) from 1992 to 2014.

Usage

data("G.counties.gdp")

Format

A data frame with 402 observations on the following 68 variables.

region_code_EU  a factor containing der EU regional code
region_code    a factor containing the German regional code
gdp1992        a numeric vector containing the GDP for German counties (Landkreise) for 1992
gdp1994        a numeric vector containing the GDP for German counties (Landkreise) for 1994
gdp1995        a numeric vector containing the GDP for German counties (Landkreise) for 1995
gdp1996        a numeric vector containing the GDP for German counties (Landkreise) for 1996
gdp1997        a numeric vector containing the GDP for German counties (Landkreise) for 1997
gdp1998        a numeric vector containing the GDP for German counties (Landkreise) for 1998
gdp1999        a numeric vector containing the GDP for German counties (Landkreise) for 1999
gdp2000        a numeric vector containing the GDP for German counties (Landkreise) for 2000
gdp2001        a numeric vector containing the GDP for German counties (Landkreise) for 2001
gdp2002        a numeric vector containing the GDP for German counties (Landkreise) for 2002
gdp2003        a numeric vector containing the GDP for German counties (Landkreise) for 2003
gdp2004        a numeric vector containing the GDP for German counties (Landkreise) for 2004
gdp2005        a numeric vector containing the GDP for German counties (Landkreise) for 2005
gdp2006        a numeric vector containing the GDP for German counties (Landkreise) for 2006
gdp2007        a numeric vector containing the GDP for German counties (Landkreise) for 2007
gdp2008        a numeric vector containing the GDP for German counties (Landkreise) for 2008
gdp2009        a numeric vector containing the GDP for German counties (Landkreise) for 2009
gdp2010        a numeric vector containing the GDP for German counties (Landkreise) for 2010
gdp2011        a numeric vector containing the GDP for German counties (Landkreise) for 2011
gdp2012        a numeric vector containing the GDP for German counties (Landkreise) for 2012
gdp2013        a numeric vector containing the GDP for German counties (Landkreise) for 2013
gdp2014 a numeric vector containing the GDP for German counties (Landkreise) for 2014
pop1992 a numeric vector containing the population for German counties (Landkreise) for 1992
pop1994 a numeric vector containing the population for German counties (Landkreise) for 1994
pop1995 a numeric vector containing the population for German counties (Landkreise) for 1995
pop1996 a numeric vector containing the population for German counties (Landkreise) for 1996
pop1997 a numeric vector containing the population for German counties (Landkreise) for 1997
pop1998 a numeric vector containing the population for German counties (Landkreise) for 1998
pop1999 a numeric vector containing the population for German counties (Landkreise) for 1999
pop2000 a numeric vector containing the population for German counties (Landkreise) for 2000
pop2001 a numeric vector containing the population for German counties (Landkreise) for 2001
pop2002 a numeric vector containing the population for German counties (Landkreise) for 2002
pop2003 a numeric vector containing the population for German counties (Landkreise) for 2003
pop2004 a numeric vector containing the population for German counties (Landkreise) for 2004
pop2005 a numeric vector containing the population for German counties (Landkreise) for 2005
pop2006 a numeric vector containing the population for German counties (Landkreise) for 2006
pop2007 a numeric vector containing the population for German counties (Landkreise) for 2007
pop2008 a numeric vector containing the population for German counties (Landkreise) for 2008
pop2009 a numeric vector containing the population for German counties (Landkreise) for 2009
pop2010 a numeric vector containing the population for German counties (Landkreise) for 2010
pop2011 a numeric vector containing the population for German counties (Landkreise) for 2011
pop2012 a numeric vector containing the population for German counties (Landkreise) for 2012
pop2013 a numeric vector containing the population for German counties (Landkreise) for 2013
pop2014 a numeric vector containing the population for German counties (Landkreise) for 2014
gdppc1992 a numeric vector containing the GDP per capita for German counties (Landkreise) for 1992
gdppc1994 a numeric vector containing the GDP per capita for German counties (Landkreise) for 1994
gdppc1995 a numeric vector containing the GDP per capita for German counties (Landkreise) for 1995
gdppc1996 a numeric vector containing the GDP per capita for German counties (Landkreise) for 1996
gdppc1997 a numeric vector containing the GDP per capita for German counties (Landkreise) for 1997
gdppc1998 a numeric vector containing the GDP per capita for German counties (Landkreise) for 1998
gdppc1999 a numeric vector containing the GDP per capita for German counties (Landkreise) for 1999
gdppc2000 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2000
gdppc2001 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2001
gdppc2002 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2002
gdppc2003 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2003
gdppc2004 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2004
gdppc2005 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2005
gdppc2006 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2006
gdppc2007 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2007
gdppc2008 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2008
gdppc2009 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2009
gdppc2010 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2010
gdppc2011 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2011
 gdppc2012 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2012
gdppc2013 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2013
gdppc2014 a numeric vector containing the GDP per capita for German counties (Landkreise) for 2014
 regional Region West or East

Details

For the years 1992 to 1999, the GDP data is incomplete.

Source


References

G.regions.emp

Examples

```r
# Regional disparities / sigma convergence in Germany
data(G.counties.gdp)
# GDP per capita for German counties (Landkreise)
cvs <- apply(G.counties.gdp[54:68], MARGIN = 2, FUN = cv)
# Calculating cv for the years 2000-2014
years <- 2000:2014
plot(years, cvs, "l", ylim=c(0.3,0.6), xlab = "year",
     ylab = "CV of GDP per capita")
# Plot cv over time
```

G.regions.emp  Employment data for German regions 2008-2014

Description

The dataset contains the industry-specific employment in the German region ("Bundeslaender") for the years 2008 to 2014.

Usage

data("G.regions.emp")

Format

A data frame with 1428 observations on the following 4 variables.

industry  a factor containing the industry (in German language, e.g. "Baugewerbe" = construction, "Handel, Gastgewerbe, Verkehr (G-I)" = retail, hospitality industry and transport industry)
region  a factor containing the names of the German regions (Bundeslaender)
year  a numeric vector containing the related year
emp  a numeric vector containing the related number of employees

Source


References

Examples

data(G.regions.emp)
# Concentration of construction industry in Germany
# based on 16 German regions (Bundeslaender) for the year 2008
construction2008 <- G.regions.emp[(G.regions.emp$industry == "Baugewerbe (F)" | G.regions.emp$industry == "Insbesamt") & G.regions.emp$year == "2008",]
# only data for construction industry (Baugewerbe) and all-over (Insbesamt)
# for the 16 German regions in the year 2008
construction2008 <- construction2008[construction2008$region != "Insbesamt",]
# delete all-over data for all industries
gini.conc(construction2008[construction2008$industry=="Baugewerbe (F)",]$emp, construction2008[construction2008$industry=="Insbesamt",]$emp)

# Concentration of financial industry in Germany 2008 vs. 2014
# based on 16 German regions (Bundeslaender) for 2008 and 2014
finance2008 <- G.regions.emp[(G.regions.emp$industry == "Erbringung von Finanz- und Vers.leistungen (K)" | G.regions.emp$industry == "Insbesamt") & G.regions.emp$year == "2008",]
finance2008 <- finance2008[finance2008$region != "Insbesamt",]
# delete all-over data for all industries
finance2014 <- G.regions.emp[(G.regions.emp$industry == "Erbringung von Finanz- und Vers.leistungen (K)" | G.regions.emp$industry == "Insbesamt") & G.regions.emp$year == "2014",]
# delete all-over data for all industries

G.regions.industries Firms and employment data for German regions 2015

Description

The dataset contains the industry-specific firm stock and employment in the German regions ("Bundeslaender") for 2015.

Usage

data("G.regions.industries")

Format

A data frame with 272 observations on the following 9 variables.

year a numeric vector containing the related year
region  a factor containing the names of the German regions (Bundeslaender)
region_code  a factor containing the codes of the German regions (Bundeslaender)
ind_code  a factor containing the codes of the industries (WZ2008)
ind_name  a factor containing the names of the industries (WZ2008)
firms  a numeric vector containing the related number of firms
emp_all  a numeric vector containing the related number of employees (incl. self-employed)
pop  a numeric vector containing the related population
area_sqkm  a numeric vector containing the related region size (in sqkm)

Source

Compiled from:
Statistisches Bundesamt (2019): Tab. 13311-0002 - Erwerbstaetige, Arbeitnehmer, Selbstaendige
und mithelfende Familienangehoerige (im Inland): Bundeslaender, Jahre, Wirtschaftszweige (Arbeitskreis "Erwerbstaetigenrechnung des Bundes und der Laender").

Examples

data(G.regions.industries)
lqs <- locq2(e_ij = G.regions.industries$emp_all,
G.regions.industries$ind_code, G.regions.industries$region_code,
LQ.output = "df")
# output as data frame
lqs_sort <- lqs[order(lqs$LQ, decreasing = TRUE),]
# Sort decreasing by size of LQ
lqs_sort[1:5,]

---

gifpro  Commercial area prognosis

Description

This function contains the basic GIFPRO model for commercial area prognosis (GIFPRO = Gewerbe-
und Industrieflaechenprognose)

Usage

gifpro(e_ij, a_i, sq_ij, rq_ij, ru_ij = NULL, ai_ij, time.base, tinterval = 1,
industry.names = NULL, output = "short")
Arguments

\( e_{ij} \)  a numeric vector with \( i \) values containing the current employment in \( i \) industries in region \( j \)

\( a_i \)  a numeric vector with \( i \) values containing the share of employees in industry \( i \) which is located in commercial areas

\( sq_{ij} \)  a numeric vector with \( i \) values containing the annual quote of resettled employees (\emph{Neuansiedlungsquote} in German) in industry \( i \), in percent

\( rq_{ij} \)  a numeric vector with \( i \) values containing the annual quote of relocated employees (\emph{Verlagerungsquote} in German) in industry \( i \), in percent

\( ru_{ij} \)  a numeric vector with \( i \) values containing the annual quote of employees in industry \( i \) which is located in reused commercial area (\emph{Wiedernutzungsquote} in German), in percent (default: \( ru_{ij} = NULL \), which represents a quote of 0 percent, meaning that no commercial area can be reused)

\( ai_{ij} \)  a numeric vector with \( i \) values containing the areal index (\emph{Flaechenkennziffer} in German), representing the area requirement (e.g. in sqm) per employee in industry \( i \)

\( \text{time.base} \)  a single value representing the start time of the prognosis (typically current year + 1)

\( t\text{interval} \)  a single value representing the forecast horizon (length of time into the future for which the commercial area prognosis is done), in time units (e.g. \( t\text{interval} = 10 \) = 10 years)

\( \text{industry.names} \)  a vector containing the industry names (e.g. from the relevant statistical classification of economic activities)

\( \text{output} \)  Type of output: \text{output} = "short" (default) shows the final number of relevant employment and commercial area requirement. If \text{output} = "full", employment and commercial area are displayed for each time unit (year)

Details

In municipal land use planning (mostly in Germany), the future need of local commercial area (which is a type of land use, defined in official land-use plans) is mostly forecasted by models founded on the GIFPRO model (\emph{Gewerbe- und Industrieflaechenbedarfsprognose}, prognosis of future demand of commercial area). GIFPRO is a demand-side model, which means predicting the demand of commercial area based on a prognosis of future employment in different industries (Bonny/Kahnert 2005). The key parameters of the model are the (assumed) shares of employees located in commercial areas \( (a_i) \), the (assumed) quotas of resettlement \( (sq_{ij}) \), relocation \( (rq_{ij}) \) and (sometimes) reuse \( (ru_{ij}) \) as well as the (assumed) area requirement per employee \( (ai_{ij}) \). Outgoing from current employment in \( i \) industries in region \( j \), \( e_{ij} \), the future employment is predicted based on the quotas mentioned above and, finally, multiplied by the industry-specific (and maybe region-specific) areal index. The GIFPRO model has been modified and extended several times, especially with respect to industry- and region-specific employment growth, quotas and areal indices (Deutsches Institut fuer Urbanistik 2010, Vallee et al. 2012).

Value

A list containing the following objects:
components
Matrices containing the single components (resettlement, relocation, reuse, relevant employment)

results
Matrices containing the final results per year and all over

Author(s)
Thomas Wieland

References


See Also
gifpro.tbs, portfolio, shift, shiftd, shifti

Examples

# Data for the city Kempten (2012):
emp2012 <- c(7228, 12452, 11589)
sharesCA <- c(100, 40, 10)
rsquote <- c(0.3, 0.3, 0.3)
rlquote <- c(0.7, 0.7, 0.7)
arealindex <- c(148, 148, 148)
industries <- c("Manufacturing", "Wholesale and retail trade, Transportation and storage, Information and communication", "Other services")

gifpro (e_ij = emp2012, a_i = sharesCA, sq_ij = rsquote, rq_ij = rlquote, ai_ij = arealindex, time.base = 2012, tinterval = 13, industry.names = industries, output = "short")
# short output

gifpro (e_ij = emp2012, a_i = sharesCA, sq_ij = rsquote, rq_ij = rlquote, ai_ij = arealindex, time.base = 2012, tinterval = 13, industry.names = industries, output = "full")
# full output

gifpro_results <- gifpro (e_ij = emp2012, a_i = sharesCA, sq_ij = rsquote, rq_ij = rlquote, ai_ij = arealindex, time.base = 2012, tinterval = 13, industry.names = industries, output = "short")
# saving results as gifpro object
gifpro.tbs

Trend-based and location-specific commercial area prognosis

Description

This function contains the TBS-GIFPRO model for commercial area prognosis (TBS-GIFPRO = Trendbasierte und standortspezifische Gewerbe- und Industrieflächenprognose; trend-based and location-specific commercial area prognosis)

Usage

gifpro.tbs(e_ij, a_i, sq_ij, rq_ij, ru_ij = NULL, ai_ij, time.base, tinterval = 1, prog.func = rep("lin", nrow(e_ij)), prog.plot = TRUE, plot.single = FALSE, multiplot.col = NULL, multiplot.row = NULL, industry.names = NULL, emp.only = FALSE, output = "short")

Arguments

e_ij
a numeric vector with \( i \) values containing the current employment in \( i \) industries in region \( j \)
a_i
a numeric vector with \( i \) values containing the share of employees in industry \( i \) which is located in commercial areas
sq_ij
a numeric vector with \( i \) values containing the annual quote of resettled employees (Neuansiedlungsquote in German) in industry \( i \), in percent
rq_ij
a numeric vector with \( i \) values containing the annual quote of relocated employees (Verlagerungsquote in German) in industry \( i \), in percent
ru_ij
a numeric vector with \( i \) values containing the annual quote of employees in industry \( i \) which is located in reused commercial area (Wiedernutzungsquote in German), in percent (default: \( ru_ij = NULL \), which represents a quote of 0 percent, meaning that no commercial area can be reused)
ai_ij
a numeric vector with \( i \) values containing the areal index (Flächenkennziffer in German), representing the area requirement (e.g. in sqm) per employee in industry \( i \)
time.base
a single value representing the start time of the prognosis (typically current year + 1)
tinterval
a single value representing the forecast horizon (length of time into the future for which the commercial area prognosis is done), in time units (e.g. \( tinterval = 10 = 10 \) years)
prog.func  a vector containing the estimation function types for employment prognosis
("lin" for linear, "pow" for power, "exp" for exponential and "logi" for logistic function); must have the same length as e_ij and industry.names, respectively.

prog.plot  Logical argument that indicates if the employment prognoses have to be plotted

plot.single  If prog.plot = TRUE: Logical argument that indicates if the plots are stored as single graphic devices or integrated in one plot

multiplot.col  No. of columns in plot

multiplot.row  No. of rows in plot

industry.names  a vector containing the industry names (e.g. from the relevant statistical classification of economic activities)

emp.only  Logical argument that indicates if the analysis only contains employment prognosis

output  Type of output: output = "short" (default) shows the final number of relevant employment and commercial area requirement. If output = "full", employment and commercial area are displayed for each time unit (year)

Details

In municipal land use planning (mostly in Germany), the future need of local commercial area (which is a type of land use, defined in official land-use plans) is mostly forecasted by models founded on the GIFPRO model (Gewerbe- und Industrieflaechenbedarfsprognose, prognosis of future demand of commercial area). GIFPRO is a demand-side model, which means predicting the demand of commercial area based on a prognosis of future employment in different industries (Bonny/Kahnert 2005). The key parameters of the model are the (assumed) shares of employees located in commercial areas (a_i), the (assumed) quotas of resettlement (sq_ij), relocation (rq_ij) and (sometimes) reuse (ru_ij) as well as the (assumed) area requirement per employee (aiij). Outgoing from current employment in i industries in region j, e_ij, the future employment is predicted based on the quotas mentioned above and, finally, multiplied by the industry-specific (and maybe region-specific) areal index. The GIFPRO model has been modified and extended several times, especially with respect to industry- and region-specific employment growth, quotas and areal indices (Deutsches Institut fuer Urbanistik 2010, Vallee et al. 2012).

This function contains the TBS-GIFPRO model for commercial area prognosis (TBS-GIFPRO = Trendbasierte und standortspezifische Gewerbe- und Industrieflaechenprognose; trend-based and location-specific commercial area prognosis) (Deutsches Institut fuer Urbanistik 2010).

Value

A list containing the following objects:

components  List with matrices containing the single components (resettlement, relocation, reuse, relevant employment)

results  List with matrices containing the final results per year and all over as well as the industry-specific forecast data

Author(s)

Thomas Wieland
References


See Also

gifpro, portfolio, shift, shiftd, shifti

Examples

# Data for Goettingen:
data(Goettingen)

anteilGOE <- rep(100,15)
nvquote <- rep(0.3, 15)
v.quote <- rep(0.7, 15)

gifpro.tbs (e_ij = Goettingen[2:16,3:12],
a_i = anteilGOE, sq_ij = nvquote,
 rq_ij = v.quote, tinterval = 12, prog.func =
 rep("lin", nrow(Goettingen[2:16,3:12])),
 ai_ij = 150, time.base = 2008, output = "full",
 industry.names = Goettingen$WZ2008_Code[2:16],
 prog.plot = TRUE, plot.single = FALSE)

gini

Description

Calculating the Gini coefficient of inequality (or concentration), standardized and non-standardized, and optionally plotting the Lorenz curve

Usage

gini(x, coefnorm = FALSE, weighting = NULL, na.rm = TRUE, lc = FALSE, lcx = "% of objects", lcy = "% of regarded variable", lctitle = "Lorenz curve", le.col = "blue", lc.col = "black", lsize = 1, ltype = "solid",

Arguments

x A numeric vector (e.g. dataset of household income, sales turnover or supply)

coefnorm logical argument that indicates if the function output is the non-standardized or
the standardized Gini coefficient (default: coefnorm = FALSE, that means the
non-standardized Gini coefficient is returned)

weighting A numeric vector containing the weighting data (e.g. size of income classes
when calculating a Gini coefficient for aggregated income data)

na.rm logical argument that indicates whether NA values should be excluded before
computing results

lc logical argument that indicates if the Lorenz curve is plotted additionally (de-
cfault: lc = FALSE, so no Lorenz curve is displayed)

lcx if lc = TRUE (plot of Lorenz curve), lcx defines the x axis label

lcy if lc = TRUE (plot of Lorenz curve), lcy defines the y axis label

lctitle if lc = TRUE (plot of Lorenz curve), lctitle defines the overall title of the
Lorenz curve plot

le.col if lc = TRUE (plot of Lorenz curve), le.col defines the color of the diagonale
(line of equality)

lc.col if lc = TRUE (plot of Lorenz curve), lc.col defines the color of the Lorenz curve

lsize if lc = TRUE (plot of Lorenz curve), lsize defines the size of the lines (default: 1)

ltype if lc = TRUE (plot of Lorenz curve), ltype defines the type of the lines (default:
"solid")

bg.col if lc = TRUE (plot of Lorenz curve), bg.col defines the background color of the
plot (default: "gray95")

bgrid if lc = TRUE (plot of Lorenz curve), the logical argument bgrid defines if a grid
is shown in the plot

bgrid.col if lc = TRUE (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.col
defines the color of the background grid (default: "white")

bgrid.size if lc = TRUE (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.size
defines the size of the background grid (default: 2)

bgrid.type if lc = TRUE (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.type
defines the type of lines of the background grid (default: "solid")

lcg if lc = TRUE (plot of Lorenz curve), the logical argument lcg defines if the non-
standardized Gini coefficient is displayed in the Lorenz curve plot

lcgn if lc = TRUE (plot of Lorenz curve), the logical argument lcgn defines if the
standardized Gini coefficient is displayed in the Lorenz curve plot
1cg.caption  if lcg = TRUE (displaying the Gini coefficient in the plot), 1cg.caption specifies the caption above the coefficients
1cg.lab.x  if lcg = TRUE (displaying the Gini coefficient in the plot), 1cg.lab.x specifies the x coordinate of the label
1cg.lab.y  if lcg = TRUE (displaying the Gini coefficient in the plot), 1cg.lab.y specifies the y coordinate of the label
add.lc  if l1c = TRUE (plot of Lorenz curve), add.lc specifies if a new Lorenz curve is plotted (add.lc = "FALSE") or the plot is added to an existing Lorenz curve plot (add.lc = "TRUE")

Details
The Gini coefficient (Gini 1912) is a popular measure of statistical dispersion, especially used for analyzing inequality or concentration. The Lorenz curve (Lorenz 1905), though developed independently, can be regarded as a graphical representation of the degree of inequality/concentration calculated by the Gini coefficient ($G$) and can also be used for additional interpretations of it. In an economic-geographical context, these methods are frequently used to analyse the concentration/inequality of income or wealth within countries (Aoyama et al. 2011). Other areas of application are analyzing regional disparities (Lessmann 2005, Nakamura 2008) and concentration in markets (sales turnover of competing firms) which makes Gini and Lorenz part of economic statistics in general (Doersam 2004, Roberts 2014).

The Gini coefficient ($G$) varies between 0 (no inequality/concentration) and 1 (complete inequality/concentration). The Lorenz curve displays the deviations of the empirical distribution from a perfectly equal distribution as the difference between two graphs (the distribution curve and a diagonal line of perfect equality). This function calculates $G$ and plots the Lorenz curve optionally. As there are several ways to calculate the Gini coefficient, this function uses the formula given in Doersam (2004). Because the maximum of $G$ is not equal to 1, also a standardized coefficient ($G^*$) with a maximum equal to 1 can be calculated alternatively. If a Gini coefficient for aggregated data (e.g. income classes with averaged incomes) or the Gini coefficient has to be weighted, use a weighting vector (e.g. size of the income classes).

Value
A single numeric value of the Gini coefficient ($0 < G < 1$) or the standardized Gini coefficient ($0 < G^* < 1$) and, optionally, a plot of the Lorenz curve.

Author(s)
Thomas Wieland

References


See Also
cv, gini.conc, gini.spec, herf, hoover

Examples

# Market concentration (example from Doersam 2004):
sales <- c(20, 50, 20, 10)
# sales turnover of four car manufacturing companies
gini(sales, lc = TRUE, lcx = "percentage of companies", lcy = "percentage of sales",
lctitle = "Lorenz curve of sales", lc = TRUE, lcgn = TRUE)
# returns the non-standardized Gini coefficient (0.3) and
# plots the Lorenz curve with user-defined title and labels
gini(sales, coefnorm = TRUE)
# returns the standardized Gini coefficient (0.4)

# Income classes (example from Doersam 2004):
income <- c(500, 1500, 2500, 4000, 7500, 15000)
# average income of 6 income classes
sizeofclass <- c(1000, 1200, 1600, 400, 200, 600)
# size of income classes
gini(income, weighting = sizeofclass)
# returns the non-standardized Gini coefficient (0.5278)

# Market concentration in automotive industry
data(Automotive)
gini(Automotive$Turnover2008, lsize=1, lc=TRUE, le.col = "black",
lc.col = "orange", lcx = "Shares of companies", lcy = "Shares of turnover / cars",
lctitle = "Automotive industry: market concentration",
lcg = TRUE, lcgn = TRUE, lcg.caption = "Turnover 2008: ", lcg.lab.x = 0, lcg.lab.y = 1)
# Gini coefficient and Lorenz curve for turnover 2008
gini(Automotive$Turnover2013, lsize=1, lc = TRUE, add.lc = TRUE, lc.col = "red",
lcg = TRUE, lcgn = TRUE, lcg.caption = "Turnover 2013: ", lcg.lab.x = 0, lcg.lab.y = 0.85)
# Adding Gini coefficient and Lorenz curve for turnover 2013
gini(Automotive$Quantity2014_car, lsize=1, lc = TRUE, add.lc = TRUE, lc.col = "blue",
lcg = TRUE, lcgn = TRUE, lcg.caption = "Cars 2014: ", lcg.lab.x = 0, lcg.lab.y = 0.7)
# Adding Gini coefficient and Lorenz curve for cars 2014
# Regional disparities in Germany:
gdp <- c(460.69, 549.19, 124.16, 65.29, 31.59, 109.27, 263.44, 39.87, 258.53,
645.59, 131.95, 35.03, 112.66, 56.22, 85.61, 56.81)
# GDP of german regions (Bundeslaender) 2015 (in billion EUR)
gini(gdp)
# returns the non-standardized Gini coefficient (0.5009)

gini.conc

Gini coefficient of spatial industry concentration

Description

Calculating the Gini coefficient of spatial industry concentration based on regional industry data (normally employment data)

Usage

gini.conc(e_ij, e_j, lc = FALSE, lcx = "% of objects",
lcx = "% of regarded variable", lctitle = "Lorenz curve",
le.col = "blue", lc.col = "black", lsize = 1, ltype = "solid",
b.g.col = "gray95", bgrid = TRUE, bgrid.col = "white",
bgrid.size = 2, bgrid.type = "solid", lc = FALSE, lcg = FALSE, lcgx = FALSE,
lg.col = NULL, lcol.lab.x = 0, lcol.lab.y = 1,
add.lc = FALSE, plot.lc = TRUE)

Arguments

e_ij a numeric vector with the employment of the industry i in region j
e_j a numeric vector with the employment in region j
lc logical argument that indicates if the Lorenz curve is plotted additionally (default: lc = FALSE, so no Lorenz curve is displayed)
lcx if lc = TRUE (plot of Lorenz curve), lcx defines the x axis label
lcy if lc = TRUE (plot of Lorenz curve), lcy defines the y axis label
lctitle if lc = TRUE (plot of Lorenz curve), lctitle defines the overall title of the Lorenz curve plot
le.col if lc = TRUE (plot of Lorenz curve), le.col defines the color of the diagonale (line of equality)
lc.col if lc = TRUE (plot of Lorenz curve), lc.col defines the color of the Lorenz curve
lsize if lc = TRUE (plot of Lorenz curve), lsize defines the size of the lines (default: 1)
ltype if lc = TRUE (plot of Lorenz curve), ltype defines the type of the lines (default: "solid")
bg.col if lc = TRUE (plot of Lorenz curve), bg.col defines the background color of the plot (default: "gray95")
bgrid if \( \text{lc} = \text{TRUE} \) (plot of Lorenz curve), the logical argument bgrid defines if a grid is shown in the plot

bgrid.col if \( \text{lc} = \text{TRUE} \) (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.col defines the color of the background grid (default: "white")

bgrid.size if \( \text{lc} = \text{TRUE} \) (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.size defines the size of the background grid (default: 2)

bgrid.type if \( \text{lc} = \text{TRUE} \) (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.type defines the type of lines of the background grid (default: "solid")

lcg if \( \text{lc} = \text{TRUE} \) (plot of Lorenz curve), the logical argument lcg defines if the non-standardized Gini coefficient is displayed in the Lorenz curve plot

lcgn if \( \text{lc} = \text{TRUE} \) (plot of Lorenz curve), the logical argument lcgn defines if the standardized Gini coefficient is displayed in the Lorenz curve plot

lcg.caption if \( \text{lcg} = \text{TRUE} \) (displaying the Gini coefficient in the plot), lcg.caption specifies the caption above the coefficients

lcg.lab.x if \( \text{lcg} = \text{TRUE} \) (displaying the Gini coefficient in the plot), lcg.lab.x specifies the x coordinate of the label

lcg.lab.y if \( \text{lcg} = \text{TRUE} \) (displaying the Gini coefficient in the plot), lcg.lab.y specifies the y coordinate of the label

add.lc if \( \text{lc} = \text{TRUE} \) (plot of Lorenz curve), add.lc specifies if a new Lorenz curve is plotted (add.lc = "FALSE") or the plot is added to an existing Lorenz curve plot (add.lc = "TRUE")

plot.lc logical argument that indicates if the Lorenz curve itself is plotted (if plot.lc = FALSE, only the line of equality is plotted))

Details

The Gini coefficient of spatial industry concentration \( (G_i) \) is a special spatial modification of the Gini coefficient of inequality (see the function gini()). It represents the rate of spatial concentration of the industry \( i \) referring to \( j \) regions (e.g. cities, counties, states). The coefficient \( G_i \) varies between 0 (perfect distribution, respectively no concentration) and 1 (complete concentration in one region). Optionally a Lorenz curve is plotted (if \( \text{lc} = \text{TRUE} \)).

Value

A single numeric value \( (0 < G_i < 1) \)

Author(s)

Thomas Wieland

References


See Also

gini.gini.spec

Examples

# Example from Farhauer/Kroell (2013):
E_ij <- c(500,500,1000,7000,1000)
# employment of the industry in five regions
E_j <- c(20000,15000,20000,40000,5000)
# employment in the five regions
gini.conc(E_ij,E_j)
# Returns the Gini coefficient of industry concentration (0.4068966)

data(G.regions.emp)
# Concentration of construction industry in Germany
# based on 16 German regions (Bundeslaender) for the year 2008
construction2008 <- G.regions.emp[(G.regions.emp$industry == "Baugewerbe (F)" |
G.regions.emp$industry == "Ins gesamt") & G.regions.emp$year == "2008",]
# only data for construction industry (Baugewerbe) and all-over (Ins gesamt)
# for the 16 German regions in the year 2008
construction2008 <- construction2008[construction2008$region != "Ins gesamt",]
# delete all-over data for all industries
gini.conc(construction2008[construction2008$industry=="Baugewerbe (F)",]$emp,
construction2008[construction2008$industry=="Ins gesamt",]$emp)

# Concentration of financial industry in Germany 2008 vs. 2014
# based on 16 German regions (Bundeslaender) for 2008 and 2014
finance2008 <- G.regions.emp[(G.regions.emp$industry ==
"Erbringung von Finanz- und Vers.leistungen (K)" |
G.regions.emp$industry == "Ins gesamt") & G.regions.emp$year == "2008",]
finance2008 <- finance2008[finance2008$region != "Ins gesamt",]
# delete all-over data for all industries
gini.conc(finance2008[finance2008$industry=="Erbringung von Finanz- und Vers.leistungen (K)",]$emp,
finance2008[finance2008$industry=="Ins gesamt",]$emp)

finance2014 <- G.regions.emp[(G.regions.emp$industry ==
"Erbringung von Finanz- und Vers.leistungen (K)" |
G.regions.emp$industry == "Ins gesamt") & G.regions.emp$year == "2014",]
finance2014 <- finance2014[finance2014$region != "Ins gesamt",]
# delete all-over data for all industries
gini.conc(finance2014[finance2014$industry=="Erbringung von Finanz- und Vers.leistungen (K)",]$emp,
finance2014[finance2014$industry=="Ins gesamt",]$emp)

---

gini.spec

Gini coefficient of regional specialization

Description

Calculating the Gini coefficient of regional specialization based on regional industry data (normally
employment data)
Usage

\texttt{gini.spec(e_ij, e_i, lc = FALSE, lcx = \texttt{"\% of objects"}, lcy = \texttt{\% of regarded variable"}, lctitle = \texttt{\"Lorenz curve"}, le.col = \texttt{\"blue\"}, lc.col = \texttt{\"black\"}, lsize = 1, ltype = \texttt{\"solid\"}, bg.col = \texttt{\"gray95\"}, bgrid = TRUE, bgrid.col = \texttt{\"white\"}, bgrid.size = 2, bgrid.type = \texttt{\"solid\"}, lcg = FALSE, lcgn = FALSE, lcg.caption = NULL, lcg.lab.x = 0, lcg.lab.y = 1, add.lc = FALSE, plot.lc = TRUE)}

Arguments

\texttt{e_ij} a numeric vector with the employment of the industries \textit{i} in region \textit{j}
\texttt{e_i} a numeric vector with the employment in the industries \textit{i}
\texttt{lc} logical argument that indicates if the Lorenz curve is plotted additionally (default: \texttt{lc = FALSE}, so no Lorenz curve is displayed)
\texttt{lcx} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{lcx} defines the x axis label
\texttt{lcy} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{lcy} defines the y axis label
\texttt{lctitle} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{lctitle} defines the overall title of the Lorenz curve plot
\texttt{le.col} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{le.col} defines the color of the diagonale (line of equality)
\texttt{lc.col} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{lc.col} defines the color of the Lorenz curve
\texttt{lsize} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{lsize} defines the size of the lines (default: 1)
\texttt{ltype} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{ltype} defines the type of the lines (default: \texttt{\"solid\"})
\texttt{bg.col} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{bg.col} defines the background color of the plot (default: \texttt{\"gray95\"})
\texttt{bgrid} if \texttt{lc = TRUE} (plot of Lorenz curve), the logical argument \texttt{bgrid} defines if a grid is shown in the plot
\texttt{bgrid.col} if \texttt{lc = TRUE} (plot of Lorenz curve) and \texttt{bgrid = TRUE} (background grid), \texttt{bgrid.col} defines the color of the background grid (default: \texttt{\"white\"})
\texttt{bgrid.size} if \texttt{lc = TRUE} (plot of Lorenz curve) and \texttt{bgrid = TRUE} (background grid), \texttt{bgrid.size} defines the size of the background grid (default: 2)
\texttt{bgrid.type} if \texttt{lc = TRUE} (plot of Lorenz curve) and \texttt{bgrid = TRUE} (background grid), \texttt{bgrid.type} defines the type of lines of the background grid (default: \texttt{\"solid\"})
\texttt{lcg} if \texttt{lc = TRUE} (plot of Lorenz curve), the logical argument \texttt{lcg} defines if the non-standardized Gini coefficient is displayed in the Lorenz curve plot
\texttt{lcgn} if \texttt{lc = TRUE} (plot of Lorenz curve), the logical argument \texttt{lcgn} defines if the standardized Gini coefficient is displayed in the Lorenz curve plot
\texttt{lcg.caption} if \texttt{lcg = TRUE} (displaying the Gini coefficient in the plot), \texttt{lcg.caption} specifies the caption above the coefficients
1cg.lab.x if lcg = TRUE (displaying the Gini coefficient in the plot), 1cg.lab.x specifies the x coordinate of the label

1cg.lab.y if lcg = TRUE (displaying the Gini coefficient in the plot), 1cg.lab.y specifies the y coordinate of the label

add.lc if lc = TRUE (plot of Lorenz curve), add.lc specifies if a new Lorenz curve is plotted (add.lc = "FALSE") or the plot is added to an existing Lorenz curve plot (add.lc = "TRUE")

plot.lc logical argument that indicates if the Lorenz curve itself is plotted (if plot.lc = FALSE, only the line of equality is plotted))

Details

The Gini coefficient of regional specialization (G_j) is a special spatial modification of the Gini coefficient of inequality (see the function gini()). It represents the degree of regional specialization of the region j referring to i industries. The coefficient G_j varies between 0 (no specialization) and 1 (complete specialization). Optionally a Lorenz curve is plotted (if lc = TRUE).

Value

A single numeric value (0 < G_j < 1)

Author(s)

Thomas Wieland

References


See Also

gini, gini.conc

Examples

# Example from Farhauer/Kroell (2013):
E_ij <- c(700,600,500,10000,40000)
# employment of five industries in the region
E_i <- c(30000,15000,10000,60000,50000)
# over-all employment in the five industries
gini.spec (E_ij, E_i)
# Returns the Gini coefficient of regional specialization (0.6222222)

# Example Freiburg
data(Freiburg)
# Loads the data
E_ij <- Freiburg$e_Freiburg2014
# industry-specific employment in Freiburg 2014
E_i <- Freiburg$e_Germany2014
# industry-specific employment in Germany 2014
gini.spec (E_ij, E_i)
# Returns the Gini coefficient of regional specialization (0.2089009)

# Example Goettingen
data(Goettingen)
# Loads the data
gini.spec(Goettingen$Goettingen2017[2:16], Goettingen$BRD2017[2:16])
# Returns the Gini coefficient of regional specialization 2017 (0.359852)

---

**gini2**

**Gini coefficient**

**Description**

Calculating the Gini coefficient of inequality (or concentration), standardized and non-standardized, and optionally plotting the Lorenz curve

**Usage**

```r
 gini2(x, weighting = NULL, coefnorm = FALSE, na.rm = TRUE)
```

**Arguments**

- `x`: A numeric vector (e.g. dataset of regional incomes)
- `weighting`: A numeric vector containing the weighting data (e.g. regional population)
- `coefnorm`: logical argument that indicates if the function output is the non-standardized or the standardized Gini coefficient (default: `coefnorm = FALSE`, which means the non-standardized Gini coefficient is returned)
- `na.rm`: logical argument that indicates whether NA values should be excluded before computing results

**Details**

The **Gini coefficient** (Gini 1912) is a popular measure of statistical dispersion, especially used for analyzing inequality or concentration. In an economic-geographical context, the Gini coefficient is frequently used to analyse the concentration/inequality of income or wealth within countries (Aoyama et al. 2011). Other areas of application are analyzing regional disparities (Lessmann 2005, Nakamura 2008) and concentration in markets (sales turnover of competing firms).

The **Gini coefficient** ($G$) varies between 0 (no inequality/concentration) and 1 (complete inequality/concentration). This function calculates $G$. As there are several ways to calculate the Gini coefficient, this function uses the formula given in Doersam (2004). Because the maximum of $G$ is not equal to 1, also a standardized coefficient ($G^*$) with a maximum equal to 1 can be calculated alternatively. If a Gini coefficient for aggregated data (e.g. income classes with averaged incomes) or the Gini coefficient has to be weighted, use a `weighting` vector (e.g. size of the income classes).
Value

A single numeric value of the Gini coefficient \((0 < G < 1)\) or the standardized Gini coefficient \((0 < G^* < 1)\) and, optionally, a plot of the Lorenz curve.

Author(s)

Thomas Wieland

References


See Also

cv, gini.conc, gini.spec, herf, hoover

Examples

# Market concentration (example from Doersam 2004):
sales <- c(20,50,20,10)
# sales turnover of four car manufacturing companies
gini(sales, lc = TRUE, lcx = "percentage of companies", lcy = "percentage of sales",
lctitle = "Lorenz curve of sales", lcg = TRUE, lcgn = TRUE)
# returns the non-standardized Gini coefficient (0.3) and
# plots the Lorenz curve with user-defined title and labels
gini(sales, coefnorm = TRUE)
# returns the standardized Gini coefficient (0.4)

# Income classes (example from Doersam 2004):
income <- c(500, 1500, 2500, 4000, 7500, 15000)
# average income of 6 income classes
sizeofclass <- c(1000, 1200, 1600, 400, 200, 600)
# size of income classes
gini (income, weighting = sizeofclass)
# returns the non-standardized Gini coefficient (0.5278)

# Market concentration in automotive industry
data(Automotive)
gini(Automotive$Turnover2008, lsize=1, lc=TRUE, le.col = "black",
lc.col = "orange", lcx = "Shares of companies", lcY = "Shares of turnover / cars",
lctitle = "Automotive industry: market concentration",
lcg = TRUE, lcgn = TRUE, lcg.caption = "Turnover 2008:", lcg.lab.x = 0, lcg.lab.y = 1)
# Gini coefficient and Lorenz curve for turnover 2008
gini(Automotive$Turnover2013, lsize=1, lc = TRUE, add.lc = TRUE, lc.col = "red",
lcg = TRUE, lcgn = TRUE, lcg.caption = "Turnover 2013:", lcg.lab.x = 0, lcg.lab.y = 0.85)
# Adding Gini coefficient and Lorenz curve for turnover 2013
gini(Automotive$Quantity2014_car, lsize=1, lc = TRUE, add.lc = TRUE, lc.col = "blue",
lcg = TRUE, lcgn = TRUE, lcg.caption = "Cars 2014:", lcg.lab.x = 0, lcg.lab.y = 0.7)
# Adding Gini coefficient and Lorenz curve for cars 2014

# Regional disparities in Germany:
gdp <- c(460.69, 549.19, 124.16, 65.29, 31.59, 109.27, 263.44, 39.87, 258.53, 645.59, 131.95, 35.03, 112.66, 56.22, 85.61, 56.81)
# GDP of German regions (Bundeslaender) 2015 (in billion EUR)
gini(gdp)
# returns the non-standardized Gini coefficient (0.5009)

---

**Employment data for Goettingen and Germany 2008-2017**

**Description**

This dataset contains the employees in 15 economic sections (German Classification of Economic Activities WZ2008) for the city Goettingen and Germany regarding the years 2008-2017 (date: 30 June each year).

**Usage**

data("Goettingen")

**Format**

A data frame with 16 observations on the following 22 variables.

- **WZ2008_Code**: a factor containing the code of the industry (15 economic sections from the German Classification of Economic Activities WZ2008 + total employees), in German language
- **WZ2008_Name**: a factor containing the name of the industry (15 economic sections from the German Classification of Economic Activities WZ2008 + total employees), in German language
- **Goettingen2008**: industry employees in the city of Goettingen 2008
Goettingen2009 industry employees in the city of Goettingen 2009
Goettingen2010 industry employees in the city of Goettingen 2010
Goettingen2011 industry employees in the city of Goettingen 2011
Goettingen2012 industry employees in the city of Goettingen 2012
Goettingen2013 industry employees in the city of Goettingen 2013
Goettingen2014 industry employees in the city of Goettingen 2014
Goettingen2015 industry employees in the city of Goettingen 2015
Goettingen2016 industry employees in the city of Goettingen 2016
Goettingen2017 industry employees in the city of Goettingen 2017
BRD2008 industry employees in Germany 2008
BRD2009 industry employees in Germany 2009
BRD2010 industry employees in Germany 2010
BRD2011 industry employees in Germany 2011
BRD2012 industry employees in Germany 2012
BRD2013 industry employees in Germany 2013
BRD2014 industry employees in Germany 2014
BRD2015 industry employees in Germany 2015
BRD2016 industry employees in Germany 2016
BRD2017 industry employees in Germany 2017

Source


References


Examples

data(Goettingen)

# Location quotients for Goettingen 2017:
locq(Goettingen$Goettingen2017[2:16], Goettingen$Goettingen2017[1],
     Goettingen$BRD2017[2:16], Goettingen$BRD2017[1])

# Gini coefficient of regional specialization 2017:
gini.spec(Goettingen$Goettingen2017[2:16], Goettingen$BRD2017[2:16])

# Krugman coefficient of regional specialization 2017:
krugman.spec(Goettingen$Goettingen2017[2:16], Goettingen$BRD2017[2:16])

GoettingenHealth1  Healthcare providers in South Lower Saxony

Description

Dataset with healthcare providers (general practitioners, psychotherapists, pharmacies) in two German counties (Goettingen and Northeim)

Usage

data("GoettingenHealth1")

Format

A data frame with 617 observations on the following 5 variables.

location  a numeric vector with unique IDs of the healthcare providers
lat       Latitude
lon       Longitude
type      Type of healthcare provider: general practitioners (phyh_gen), psychotherapists (psych) or pharmacies (pharm)
district  a numeric vector containing the IDs of the district the specific provider is located in

Source


References

## Not run:
```
data(GoettingenHealth1)
data(GoettingenHealth1)
data(GoettingenHealth1)

area_goe <- 1753000000
area_nom <- 1267000000

area_gn <- area_goe+area_nom
sqrt(area_gn/pi)

ripley(GoettingenHealth1[GoettingenHealth1$type == "phys_gen",]
       "location", "lat", "lon", area = area_gn, t.max = 30000, t.sep = 300)

ripley(GoettingenHealth1[GoettingenHealth1$type == "pharm",]
       "location", "lat", "lon", area = area_gn, t.max = 30000, t.sep = 300)

ripley(GoettingenHealth1[GoettingenHealth1$type == "psych",]
       "location", "lat", "lon", area = area_gn, t.max = 30000, t.sep = 300)
```

## End(Not run)

---

GoettingenHealth2  Healthcare provision in South Lower Saxony

### Description

Dataset with districts in two German counties (Goettingen and Northeim) and the corresponding healthcare providers (general practitioners, psychotherapists, pharmacies) and population size.

### Usage

```
data("GoettingenHealth2")
```

### Format

A data frame with 420 observations on the following 7 variables.

- **district**: a numeric vector containing the IDs of the district
- **pop**: no. of inhabitants
- **lat**: Latitude
- **lon**: Longitude
- **phys_gen**: no. of general practitioners
- **psych**: no. of psychotherapists
- **pharm**: no. of pharmacies
growth

Source

References

Examples
data(GoettingenHealth2)
# districts with healthcare providers and population size
williamson((GoettingenHealth2$phys_gen/GoettingenHealth2$pop), GoettingenHealth2$pop)

growth

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>This function calculates the growth from two input numeric vectors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>growth(val1, val2, growth.type = &quot;growth&quot;, output = &quot;rate&quot;, rate.perc = FALSE, log.rate = FALSE, factor.mean = &quot;mean&quot;, time.periods = NULL)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>val1</td>
</tr>
<tr>
<td>val2</td>
</tr>
<tr>
<td>growth.type</td>
</tr>
<tr>
<td>output</td>
</tr>
<tr>
<td>rate.perc</td>
</tr>
<tr>
<td>log.rate</td>
</tr>
<tr>
<td>factor.mean</td>
</tr>
<tr>
<td>time.periods</td>
</tr>
</tbody>
</table>
Value

A numeric vector containing the growth rates in the same order as stated

Author(s)

Thomas Wieland

Examples

# Example from Farhauer/Kroell (2013):
region_A_t <- c(90,20,10,60)
region_A_t1 <- c(100,40,10,55)
# data for region A (time t and t+1)
nation_X_t <- c(400,150,150,400)
nation_X_t1 <- c(440,210,135,480)
# data for the national economy (time t and t+1)
growth(region_A_t, region_A_t1)

data(Freiburg)
# Loads the data
growth(Freiburg$e_Freiburg2008, Freiburg$e_Freiburg2014, growth.type = "rate")
# Industry-specific growth rates for Freiburg 2008 to 2014

hansen

Hansen accessibility

Description

Calculating the Hansen accessibility for given origins and destinations

Usage

hansen(od_dataset, origins, destinations, attrac, dist, gamma = 1, lambda = -2,
atype = "pow", dtype = "pow", gamma2 = NULL, lambda2 = NULL, dist_const = 0,
dist_max = NULL, extract_local = FALSE, accnorm = FALSE, check_df = TRUE,
print.results = TRUE)

Arguments

od_dataset an interaction matrix which is a data.frame containing the origins, destinations, the distances between them and a size variable for the opportunities of the destinations

origins the column in the interaction matrix od_dataset containing the origins

destinations the column in the interaction matrix od_dataset containing the destinations

attrac the column in the interaction matrix od_dataset containing the "attractivity" variable of the destinations (e.g. no. of opportunities)
The column in the interaction matrix `od_dataset` containing the transport costs (e.g., travelling time, distance)

a single numeric value for the exponential weighting ($\gamma$) of size (default: 1)

A single numeric value for the exponential weighting ($\lambda$) of distance (transport costs, default: -2)

Type of attractivity weighting function: `atype` = "pow" (power function), `atype` = "exp" (exponential function) or `atype` = "logistic" (default: `atype` = "pow")

Type of distance weighting function: `dtype` = "pow" (power function), `dtype` = "exp" (exponential function) or `dtype` = "logistic" (default: `dtype` = "pow")

If `atype` = "logistic" a second $\gamma$ parameter is needed

If `dtype` = "logistic" a second $\lambda$ parameter is needed

A numeric value of a constant to be added to the transport costs (e.g., 1)

A numeric value of a maximal value of transport costs for the opportunities to be recognized

Logical argument that indicates if the start points should be included in the analysis or not (if $i = j$). Default value: `extract_local` = FALSE

Logical argument that indicates if the Hansen accessibility should be standardized

Logical argument that indicates if the given dataset is checked for correct input, only for internal use, should not be deselected (default: TRUE)

Logical argument that indicates if the results are shown (default: TRUE)

Accessibility and the inhibiting effect of transport costs on spatial interactions belong to the key concepts of economic geography (Aoyama et al. 2011). The Hansen accessibility (Hansen 1959) can be regarded as a potential model of spatial interaction that describes accessibility as the sum of all opportunities $O$ in the regions $j$, $O_j$, weighted by distance or other types of transport costs from the origins, $i$, to them, $d_{ij}$: $A_i = \sum_j O_j f(d_{ij})$. The distance/travel time is weighted by a distance decay function ($f(d_{ij})$) to reflect the disutility (opportunity costs) of distance. From a microeconomic perspective, the accessibility of a region or zone can be seen as the sum of all utilities of every opportunity outgoing from given starting points, given an utility function containing the opportunities (utility) and transport costs (disutility) (Orpana/Lampinen 2003). As the accessibility model originally comes from urban land use theory, it can also be used to model spatial concentration/agglomeration, e.g., to quantify the rate of agglomeration of retail locations (Orpana/Lampinen 2003, Wieland 2015).

Originally the weighting function of distance is not explicitly stated and the "attractivities" (e.g., size of the activity at the destinations) is not weighted. These specifications are relaxed in this function, so both variables can be weighted by a power, exponential or logistic function. If `accnorm` = TRUE, the Hansen accessibility is standardized by weighting the non-standardized values by the sum of all opportunities without regarding transport costs; the standardized Hansen accessibility has a range between 0 and 1.
Value

A list containing the following objects:

- origins: A data frame containing the origins
- accessibility: A data frame containing the calculated accessibility values (optional: standardized accessibilities)

Author(s)

Thomas Wieland

References


See Also

converse, dist.calc, dist.mat, dist.buf, huff, reilly

Examples

# Example from Levy/Weitz (2009):
# Data for the existing and the new location
locations <- c("Existing Store", "New Store")
S_j <- c(5000, 10000)
location_data <- data.frame(locations, S_j)
# Data for the two communities (Rock Creek and Oak Hammock)
communities <- c("Rock Creek", "Oak Hammock")
C_i <- c(5000000, 3000000)
community_data <- data.frame(communities, C_i)
# Combining location and submarket data in the interaction matrix
interactionmatrix <- merge (community_data, location_data)
# Adding driving time:
interactionmatrix[1,5] <- 10
interactionmatrix[2,5] <- 5
interactionmatrix[3,5] <- 5
interactionmatrix[4,5] <- 15
colnames(interactionmatrix) <- c("communities", "C_i", "locations", "S_j", "d_ij")
shoppingcenters1 <- interactionmatrix
huff_shares <- huff(shoppingcenters1, "communities", "locations", "S_j", "d_ij")
# Market shares of the new location:
huff_shares$ijmatrix[huff_shares$ijmatrix$locations == "New Store",]
# Hansen accessibility for Oak Hammock and Rock Creek:
# hansen (huff_shares$ijmatrix, "communities", "locations", "S_j", "d_ij")

---

**herf**  
*Herfindahl-Hirschman coefficient*

**Description**
Calculating the Herfindahl-Hirschman coefficient of concentration, standardized and non-standardized

**Usage**
herf(x, coefnorm = FALSE, output = "HHI", na.rm = TRUE)

**Arguments**
- *x* A numeric vector (e.g. dataset of sales turnover or size of firms)
- *coefnorm* logical argument that indicates if the function output is the non-standardized or the standardized Herfindahl-Hirschman coefficient (default: coefnorm = FALSE, that means the non-standardized Herfindahl-Hirschman coefficient is returned)
- *output* argument to state the output. If output = "HHI" (default), the Herfindahl-Hirschman coefficient is returned (standardized or non-standardized). If output = "eq", the Herfindahl-Hirschman coefficient equivalent number is returned
- *na.rm* logical argument that indicates whether NA values should be excluded before computing results

**Details**
The *Herfindahl-Hirschman coefficient* is a popular measure of statistical dispersion, especially used for analyzing concentration in markets, regarding sales turnovers or sizes of *n* competing firms in an industry. This indicator is especially used as a measure of market power and distortions of competition in the governmental competition policy (Roberts 2014). But the coefficient is also utilized as a measure of geographic concentration of industries (Lessmann 2005, Nakamura/Morrison Paul 2009).

The coefficient (*HHI*) varies between $\frac{1}{n}$ (parity resp. no concentration) and 1 (complete concentration). Because the minimum of *HHI* is not equal to 0, also a standardized coefficient (*HHI*+) with a minimum equal to 0 can be calculated alternatively. The *equivalent number* (which is the inverse of the *Herfindahl-Hirschman coefficient*) reflects the theoretical number of economic objects (normally firms) where a calculated coefficient is $\frac{1}{n}$, which means parity (Doersam 2004). In a regional context, the inverse of HHI is also used as a measure of diversity (Duranton/Puga 2000).
Value

A single numeric value of the Herfindahl-Hirschman coefficient \( \frac{1}{n} < HHI < 1 \) or the standardized Herfindahl-Hirschman coefficient \( 0 < HHI^* < 1 \) or the Herfindahl-Hirschman coefficient equivalent number \( H_{eq} \geq 1 \).

Author(s)

Thomas Wieland

References


See Also

cv, gini

Examples

# Example from Doersam (2004):
sales <- c(20,50,20,10)
# sales turnover of four car manufacturing companies
herf(sales)
# returns the non-standardized HHI (0.34)
herf(sales, coefnorm=TRUE)
# returns the standardized HHI (0.12)
herf(sales, output = "eq")
# returns the HHI equivalent number (2.94)

# Regional disparities in Germany:
gdp <- c(460.69, 549.19, 124.16, 65.29, 31.59, 109.27, 263.44, 39.87, 258.53, 645.59, 131.95, 35.03, 112.66, 56.22, 85.61, 56.81)
# GDP of german regions 2015 (in billion EUR)
herf(gdp)
# returns the HHI (0.125)
### Description
Calculating the Hoover Concentration Index with respect to regional income (e.g. GDP) and population.

### Usage

```r
hoover(x, ref = NULL, weighting = NULL, output = "HC", na.rm = TRUE)
```

### Arguments
- **x**: A numeric vector (dataset of regional income, e.g. GDP)
- **ref**: A numeric vector containing the reference distribution for the Hoover Index, e.g. population. If `ref = NULL`, the reference distribution is set to $1/n$.
- **weighting**: A numeric containing the weightings for the Hoover Index, e.g. population.
- **output**: Default option is the output of the Hoover Index. If `output = "data"`, the corresponding data table is returned instead.
- **na.rm**: logical argument that indicates whether NA values should be excluded before computing results.

### Details
The Hoover Concentration Index ($CI$) measures the economic concentration of income across space by comparing the share of income (e.g. GDP - Gross Domestic Product) with the share of population. The index varies between 0 (no inequality/concentration) and 1 (complete inequality/concentration). It can be used for economic inequality and/or regional disparities (Huang/Leung 2009).

### Value
A single numeric value of the Hoover Concentration Index ($0 < CI < 1$).

### Author(s)
Thomas Wieland

### References
See Also

cv.gini, herf.theil, atkinson.coulter, disp

Examples

# Regional disparities in Germany:
gdp <- c(460.69, 549.19, 124.16, 65.29, 31.59, 109.27, 263.44, 39.87, 258.53,
645.59, 131.95, 35.83, 112.66, 56.22, 85.61, 56.81)
# GDP of german regions 2015 (in billion EUR)
pop <- c(10879618, 12843514, 3520031, 2484826, 671489, 1787408, 6176172,
1612362, 7926599, 17865516, 4052803, 995597, 4084851, 2245470, 2858714, 2170714)
# population of german regions 2015
hoover(gdp, pop)

howard.cl

*Howard-Newman-Tarp colocation index*

Description

Calculating the colocation index (CL) by Howard, Newman and Tarp for two industries

Usage

howard.cl(k, industry, region, industry1, industry2, e_k = NULL)

Arguments

- **k** a vector containing the IDs/names of firms
- **industry** a vector containing the IDs/names of the industries
- **region** a vector containing the IDs/names of the regions
- **industry1** Regarded industry 1 (out of the industry vector)
- **industry2** Regarded industry 2 (out of the industry vector)
- **e_k** Employment of firm

Details

The Howard-Newman-Tarp colocation index (CL) is standardized (0 ≤ CL ≤ 1). Processing time depends on the number of firms.

Value

A single value of CL

Author(s)

Thomas Wieland
howard.xcl

References


See Also

howard.xcl, howard.xcl2, ellison.c, ellison.c2

Examples

# example from Howard et al. (2016):
firms <- 1:6
locations <- c("X", "X", "X", "Y", "Y", "X")

howard.cl(firms, industries, locations, industry1 = "A", industry2 = "B")

howard.xcl

Howard-Newman-Tarp excess colocation (XCL) index

Description

Calculating the excess colocation (XCL) index by Howard, Newman and Tarp for two industries

Usage

howard.xcl(k, industry, region, industry1, industry2, no.samples = 50, e_k = NULL)

Arguments

k a vector containing the IDs/names of firms k
industry a vector containing the IDs/names of the industries i
region a vector containing the IDs/names of the regions j
industry1 Regarded industry 1 (out of the industry vector)
industry2 Regarded industry 2 (out of the industry vector)
no.samples Number of samples for the counterfactual firm allocation via bootstrapping
e_k Employment of firm k

Details

The Howard-Newman-Tarp excess colocation index (XCL) is standardized $(-1 \leq C L \leq 1)$. The rationale behind is that the CL index (see howard.cl) is compared to a counterfactual (random) location pattern which is constructed via bootstrapping. Processing time depends on the number of firms and the number of samples.
Value

A single value of XCL

Author(s)

Thomas Wieland

References


See Also

howard.cl, howard.xcl2, ellison.c, ellison.c2

Examples

# example from Howard et al. (2016):
firms <- 1:6
locations <- c("X", "X", "X", "Y", "Y", "X")

howard.xcl(firms, industries, locations, industry1 = "A", industry2 = "B")

howard.xcl2

Howard-Newman-Tarp excess colocation (XCL) index

Description

Calculating the excess colocation (XCL) index by Howard, Newman and Tarp for a given number of industries

Usage

howard.xcl2(k, industry, region, print.results = TRUE)

Arguments

k a vector containing the IDs/names of firms k
industry a vector containing the IDs/names of the industries i
region a vector containing the IDs/names of the regions j
print.results logical argument that indicates whether the calculated values are printed or not
Details

The Howard-Newman-Tarp excess colocation index \((XCL)\) is standardized \((-1 \leq CL \leq 1)\). The rationale behind is that the CL index (see \texttt{howard.cl}) is compared to a counterfactual (random) location pattern which is constructed via bootstrapping. Processing time depends on the number of firms and the number of samples. This function takes a while even for a relatively small number of industries!

Value

A matrix with \(I\) rows (one for each industry-industry combination) containing the \(XCL\) values

Author(s)

Thomas Wieland

References


See Also

\texttt{howard.cl, howard.xcl2, ellison.c, ellison.c2}

Examples

```r
## Not run:
# example data from Farhauer/Kroell (2014):
data (FK2014_EGC)

howard.xcl2 (FK2014_EGC$firm, FK2014_EGC$industry,
FK2014_EGC$region)
# this may take a while!

## End(Not run)
```

\texttt{huff} \hspace{1cm} \textit{Huff model}

Description

Calculating market areas using the probabilistic market area model by Huff

Usage

\texttt{huff(huffdataset, origins, locations, attrac, dist, gamma = 1, lambda = -2, atype = "pow", dtype = "pow", gamma2 = NULL, lambda2 = NULL, localmarket_dataset = NULL, origin_id = NULL, localmarket = NULL, check_df = TRUE)}
Arguments

huffdataset  an interaction matrix which is a data.frame containing the origins, locations and the explanatory variables
origins  the column in the interaction matrix huffdataset containing the origins (e.g. ZIP codes)
locations  the column in the interaction matrix huffdataset containing the locations (e.g. store codes)
attrac  the column in the interaction matrix huffdataset containing the attractiveness variable (e.g. sales area)
dist  the column in the interaction matrix huffdataset containing the transport costs (e.g. travelling time)
gamma  a single numeric value for the exponential weighting of size (default: 1)
lambda  a single numeric value for the exponential weighting of distance (transport costs, default: -.2)
atype  Type of attractiveness weighting function: atype = "pow" (power function), atype = "exp" (exponential function) or atype = "logistic" (default: atype = "pow")
dtype  Type of distance weighting function: dtype = "pow" (power function), dtype = "exp" (exponential function) or dtype = "logistic" (default: dtype = "pow")
gamma2  if atype = "logistic" a second $\gamma$ parameter is needed
lambda2  if dtype = "logistic" a second $\lambda$ parameter is needed
localmarket_dataset  if output = "total", a data.frame is needed which contains data about the origins
origin_id  the ID variable of the origins in localmarket_dataset
localmarket  the customer/purchasing power potential of the origins in localmarket_dataset
check_df  logical argument that indicates if the given dataset is checked for correct input, only for internal use, should not be deselected (default: TRUE)

Details

The Huff Model (Huff 1962, 1963, 1964) is the most popular spatial interaction model for retailing and services and belongs to the family of probabilistic market area models. The basic idea of the model is that consumer decisions are not deterministic but probabilistic, so the decision of customers for a shopping location in a competitive environment cannot be predicted exactly. The results of the model are probabilities for these decisions, which can be interpreted as market shares of the regarded locations ($j$) in the customer origins ($i$), $p_{ij}$, which can be regarded as an equilibrium solution with logically consistent market shares ($0 < p_{ij} < 1, \sum_{j=1}^{n} p_{ij} = 1$). From a theoretical perspective, the model is based on an utility function with two explanatory variables ("attractivity" of the locations, transport costs between origins and locations), which are weighted by an exponent: $U_{ij} = A^\gamma d_{ij}^{-\lambda}$. This specification is relaxed in this case, so both variables can be weighted by a power, exponential or logistic function.

This function computes the market shares from a given interaction matrix and given weighting parameters. The function returns an estimated interaction matrix. If local market information about the origins (e.g. purchasing power, population size etc.) is stated, the location total turnovers are filed in another data.frame. Note that each attractiveness or distance value must be greater than zero.
Value

A list containing the following objects:

huffmat A data frame containing the Huff interaction matrix
totals If total turnovers are estimated: a data frame containing the total values (turnovers) of each location

Note

This function contains code from the authors’ package MCI.

Author(s)

Thomas Wieland

References


See Also

cconverse, reilly, hansen

Examples

# Example from Levy/Weitz (2009):

# Data for the existing and the new location
locations <- c("Existing Store", "New Store")
S_j <- c(5000, 10000)
location_data <- data.frame(locations, S_j)
# Data for the two communities (Rock Creek and Oak Hammock)
communities <- c("Rock Creek", "Oak Hammock")
C_i <- c(5000000, 3000000)
community_data <- data.frame(communities, C_i)

# Combining location and submarket data in the interaction matrix
interactionmatrix <- merge (communities, location_data)
# Adding driving time:
interactionmatrix[1,4] <- 10
interactionmatrix[2,4] <- 5
interactionmatrix[3,4] <- 5
interactionmatrix[4,4] <- 15
colnames(interactionmatrix) <- c("communities", "locations", "S_j", "d_ij")

huff_shares <- huff(interactionmatrix, "communities", "locations", "S_j", "d_ij")

# Market shares of the new location:
huff_shares$ijmatrix[huff_shares$ijmatrix$locations == "New Store",]

huff_all <- huff(interactionmatrix, "communities", "locations", "S_j", "d_ij",
localmarket_dataset = community_data, origin_id = "communities", localmarket = "C_i")

huff_all

huff_all$totals

---

**krugman.conc**  
*Krugman coefficient of spatial industry concentration for two industries*

**Description**
Calculating the Krugman coefficient for the spatial concentration of two industries based on regional industry data (normally employment data)

**Usage**

```
krugman.conc(e_ij, e_uj)
```

**Arguments**

- `e_ij` a numeric vector with the employment of the industry `i` in regions `j`
- `e_uj` a numeric vector with the employment of the industry `u` in region `j`
Details

The Krugman coefficient of industry concentration \((K_{iu})\) is a measure for the dissimilarity of the spatial structure of two industries \((i\) and \(u\)) regarding the employment in the \(j\) regions. The coefficient \(K_{iu}\) varies between 0 (no concentration/same structure) and 2 (maximum difference, that means a complete other spatial structure of the industry compared to the others). The calculation is based on the formulae in Farhauer/Kroell (2013).

Value

A single numeric value \((0 < K_{iu} < 2)\)

Author(s)

Thomas Wieland

References


See Also

gini.conc, gini.spec, krugman.conc2, krugman.spec, krugman.spec2, locq

Examples

\[
E_{ij} <- c(4388, 37489, 129423, 60941)
E_{uj} <- E_{ij}/2
krugman.conc(E_{ij}, E_{uj})
# exactly the same structure (= no concentration)
\]

krugman.conc2

Krugman coefficient of spatial industry concentration for more than two industries

Description

Calculating the Krugman coefficient for the spatial concentration of an industry based on regional industry data (normally employment data) compared with a vector of other industries

Usage

krugman.conc2(e_ij, e_uj)
Arguments

- `e_ij`  
a numeric vector with the employment of the industry \(i\) in regions \(j\)

- `e_uj`  
a data frame with the employment of the industry \(u\) in \(j\) regions

Details

The **Krugman coefficient of industry concentration** \((K_i)\) is a measure for the dissimilarity of the spatial structure of one industry \((i)\) compared to several others \((u)\) regarding the employment in the \(j\) regions. The coefficient \(K_{iu}\) varies between 0 (no concentration/same structure) and 2 (maximum difference, that means a complete other spatial structure of the industry compared to the others). The calculation is based on the formulae in Farhauer/Kroell (2013).

Value

A single numeric value \((0 < K_i < 2)\)

Author(s)

Thomas Wieland

References


See Also

gini.conc, gini.spec, krugman.conc, krugman.spec, krugman.spec2, locq

Examples

```r
# Example from Farhauer/Kroell (2013):
Chemie <- c(20000,11000,31000,8000,20000)
Sozialwesen <- c(40000,10000,25000,9000,16000)
Elektronik <- c(10000,11000,14000,14000,13000)
Holz <- c(7000,7500,11000,1500,36000)
Bergbau <- c(4320, 7811, 3900, 2300, 47560)
# five industries
industries <- data.frame(Chemie, Sozialwesen, Elektronik, Holz)
# data frame with all comparison industries
krugman.conc2(Bergbau, industries)
# returns the Krugman coefficient for the concentration
# of the mining industry (Bergbau) compared to
# chemistry (Chemie), social services (Sozialwesen),
# electronics (Elektronik) and wood industry (Holz)
# 0.8619
```
Krugman coefficient of regional specialization for two regions

Description

Calculating the Krugman coefficient for the specialization of two regions based on regional industry data (normally employment data).

Usage

```r
krugman.spec(e_ij, e_il)
```

Arguments

- `e_ij`: a numeric vector with the employment of the industries $i$ in region $j$
- `e_il`: a numeric vector with the employment of the industries $i$ in region $l$

Details

The *Krugman coefficient of regional specialization* ($K_{jl}$) is a measure for the dissimilarity of the industrial structure of two regions ($j$ and $l$) regarding the employment in the $i$ industries in these regions. The coefficient $K_{jl}$ varies between 0 (no specialization/same structure) and 2 (maximum difference, that means there is no single industry localized in both regions). The calculation is based on the formulae in Farhauer/Kroell (2013).

Value

A single numeric value ($0 < K_{jl} < 2$)

Author(s)

Thomas Wieland

References


See Also

gini.conc, gini.spec, krugman.conc, krugman.conc2, krugman.spec2, locq
Examples

# Example from Farhauer/Kroell (2013), modified:
E_ij <- c(20, 10, 70, 0, 0)
# employment of five industries in region j
E_il <- c(0, 0, 0, 60, 40)
# employment of five industries in region l
krugman.spec(E_ij, E_il)
# results the specialization coefficient (2)

# Example Goettingen:
data(Goettingen)
krugman.spec(Goettingen$Goettingen2017[2:16], Goettingen$BRD2017[2:16])
# Returns the Krugman coefficient of regional specialization 2017 (0.4508469)

krugman.spec2
Krugman coefficient of regional specialization for more than two regions

Description

Calculating the Krugman coefficient for the specialization of one region based on regional industry data (normally employment data) compared with a vector of other regions

Usage

krugman.spec2(e_ij, e_il)

Arguments

e_ij a numeric vector with the employment of the industries \( i \) in region \( j \)

e_il a data frame with the employment of the industries \( i \) in \( l \) regions

Details

The Krugman coefficient of regional specialization \((K_{jl})\) is a measure for the dissimilarity of the industrial structure of regions \((j\) and other regions, \(l\)) regarding the employment in the \(i\) industries in these regions. The coefficient \(K_{jl}\) varies between 0 (no specialization/same structure) and 2 (maximum difference, that means there is no single industry localized in both regions).

Value

A single numeric value \((0 < K_{jl} < 2)\)

Author(s)

Thomas Wieland
References


See Also

gini.conc, gini.spec, krugman.spec, krugman.conc, krugman.conc2, locq

Examples

# Example from Farhauer/Kroell (2013):
Sweden <- c(45000, 15000, 32000, 10000, 30000)
Norway <- c(35000, 12000, 30000, 8000, 22000)
Denmark <- c(40000, 10000, 25000, 9000, 18000)
Finland <- c(30000, 11000, 18000, 3000, 13000)
Island <- c(40000, 6000, 11000, 2000, 12000)
# industry jobs in five industries for five countries
countries <- data.frame(Norway, Denmark, Finland, Island)
# data frame with all comparison countries
krugman.spec2(Sweden, countries)
# returns the Krugman coefficient for the specialization
# of sweden compared to Norway, Denmark, Finland and Island
# 0.1595

litzenberger

- Littenberger-Sternberg Cluster Index

Description

Calculating the Cluster Index by Litzenberger and Sternberg

Usage

litzenberger(e_ij, e_i, a_j, a, p_j, p, b_ij, b_i)

Arguments

e_ij a single numeric value with the employment of industry i in region j
e_i a single numeric value with the over-all employment in industry i
a_j a single numeric value of the area of region j
a a single numeric value of the total area
p_j a single numeric value of the population of region j
p a single numeric value of the total population
b_ij a single numeric value of the number of firms of industry i in region j
b_i a single numeric value of the total number of firms of industry i
Details

The Litzenberger-Sternberg Cluster Index is not standardized and depends on the number of regarded industries and regions.

Value

A single numeric value of \((CI)\).

Author(s)

Thomas Wieland

References


See Also

litzenberger2, gini.conc, gini.spec, locq, locq2, ellison.a, ellison.a2, ellison.c, ellison.c2

Examples

# Example from Farhauer/Kroell (2014):
litzenberger(e_ij = 1743, e_i = 5740, a_j = 50, a = 576, p_j = 488, p = 4621, b_ij = 35, b_i = 53)
# 21.87491

<table>
<thead>
<tr>
<th>litzenberger2</th>
<th>Litzenberger-Sternberg Cluster Index</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

Calculating the Cluster Index by Litzenberger and Sternberg for a given number of \(I\) industries and \(J\) regions

Usage

litzenberger2(e_ij, industry.id, region.id, a_j, p_j, b_ij, CI.output = "mat", na.rm = TRUE)
Arguments

\[
e_{ij} \quad \text{a vector with the employment of industry } i \text{ in region } j
\]
\[
\text{industry.id} \quad \text{a vector containing the IDs of the industries } i
\]
\[
\text{region.id} \quad \text{a vector containing the IDs of the regions } j
\]
\[
a_j \quad \text{a vector containing the areas of the regions } j
\]
\[
p_j \quad \text{a vector containing the populations of the regions } j
\]
\[
b_{ij} \quad \text{a vector containing the numbers of firms of industry } i \text{ in region } j
\]
\[
\text{CI.output} \quad \text{Type of output: matrix (default: CI.output = "mat") or data frame (CI.output = "df")}
\]
\[
\text{na.rm} \quad \text{logical argument that indicates whether NA values should be excluded before computing results}
\]

Details

The Litzenberger-Sternberg Cluster Index is not standardized and depends on the number of regarded industries and regions.

Value

A matrix or data frame containing \(I \times J\) values of \(CI\)

Author(s)

Thomas Wieland

References


See Also

litzenberger, gini.conc, gini.spec, locq, locq2, ellison.a, ellison.a2, ellison.c, ellison.c2

Examples

data (G.regions.industries)

lss <- litzenberger2(G.regions.industries$emp_all, G.regions.industries$ind_code, G.regions.industries$region_code, G.regions.industries$area_sqkm, G.regions.industries$pop, G.regions.industries$firms, CI.output = "df")
# output as data frame
lss_sort <- lss[order(lss$CI, decreasing = TRUE),]
# Sort decreasing by size of CI
lss_sort[1:5,]

## lm.beta

### Beta regression coefficients

**Description**

Calculating the standardized (beta) regression coefficients of linear models

**Usage**

lm.beta(linmod, dummy.na = TRUE)

**Arguments**

- **linmod**: A `lm` object (linear regression model) with more than one independent variable
- **dummy.na**: logical argument that indicates if dummy variables should be ignored when calculating the beta weights (default: TRUE). Note that beta weights of dummy variables do not make any sense

**Details**

Standardized coefficients (beta coefficients) show how many standard deviations a dependent variable will change when the regarded independent variable is increased by a standard deviation. The β values are used in multiple linear regression models to compare the real effect (power) of the independent variables when they are measured in different units. Note that β values do not make any sense for dummy variables since they cannot change by a standard deviation.

**Value**

A list containing all independent variables and the corresponding standardized coefficients.

**Author(s)**

Thomas Wieland

**References**

Examples

x1 <- runif(100)
x2 <- runif(100)
# random values for two independent variables (x1, x2)
y <- runif(100)
# random values for the dependent variable (y)
testmodel <- lm(y~x1+x2)
# OLS regression
summary(testmodel)
# summary
lm.beta(testmodel)
# beta coefficients

locq

Location quotient

Description

Calculating the location quotient (a.k.a. Hoover-Balassa quotient)

Usage

locq(e_ij, e_j, e_i, e, industry.names = NULL, plot.results = FALSE,
LQ.method = "m", plot.title = "Localization quotients",
bar.col = "lightblue", line.col = "red", arg.size = 1)

Arguments

e_ij a single numeric value or vector with the employment of industry/industries i in region j
e_j a single numeric value with the over-all employment in region j
e_i a single numeric value or vector with the over-all employment in industry/industries i
e a single numeric value with the over-all employment in all regions
industry.names Industry names (e.g. from the relevant statistical classification of economic activities)
plot.results Logical argument that indicates if the results have to be plotted (only available if i > 1)
LQ.method Indicates whether the multiplicative (default: LQ.method = "m") or the additive LQ (LQ.method = "m") is computed
plot.title If plot.results = TRUE: Plot title
bar.col If plot.results = TRUE: Bar colour
line.col If plot.results = TRUE: LQ1-line colour
arg.size If plot.results = TRUE: Size of industry names in bar plot
Details

The location quotient is a simple measure for the concentration of an industry \((i)\) in a region \((j)\) and is also the mathematical basis for other related indicators in regional economics (e.g. \texttt{gini.conc()}). The function returns the value \(LQ\) which is equal to 1 if the concentration of the regarded industry is exactly the same as the over-all concentration (that means, it is proportionally represented in region \(j\)). If the value of \(LQ\) is smaller (bigger) than 1, the industry is underrepresented (overrepresented). The function checks the input values for errors (i.e. if employment in a region is bigger than over-all employment).

Value

A single numeric value of \((LQ)\) or a matrix with respect to all \(i\) industries. Optional: plot.

Author(s)

Thomas Wieland

References


See Also

\texttt{gini.conc, gini.spec, locq2}

Examples

# Example from Farhauer/Kroell (2013):
\texttt{locq (1714, 79006, 879213, 15593224)}
# returns the location quotient (0.3847623)

# Location quotients for Goettingen 2017:
data(Goettingen)
\texttt{locq (Goettingen$Goettingen2017[2:16], Goettingen$Goettingen2017[1], Goettingen$BRD2017[2:16], Goettingen$BRD2017[1])}
locq.growth  

Portfolio matrix for specialization and growth

Description

Portfolio matrix plot comparing two numeric vectors (here: specialization and growth)

Usage

locq.growth(e_ij1, e_ij2, e_i1, e_i2, industry.names = NULL, 
y.axis = "r", 
psize, psizen.factor = 10, time.periods = NULL, 
pxm = "Regional specialization", pmy = "Regional growth", 
ptitle = "Portfolio matrix", pcol = NULL, pcol.border = NULL, 
leg = FALSE, leg.fsize = 1, leg.col = NULL, 
bg.col = "gray95", bgrid = TRUE, bgrid.col = "white", 
bgird.size = 2, bgrid.type = "solid", 
seg.x = 1, seg.y = 0)

Arguments

e_ij1  a numeric vector with i values containing the employment in i industries in region j at time 1

e_ij2  a numeric vector with i values containing the employment in i industries in region j at time 2

e_1  a numeric vector with i values containing the total employment in i industries at time 1

e_i2  a numeric vector with i values containing the total employment in i industries at time 2
industry.names  Industry names (e.g. from the relevant statistical classification of economic activities)
y.axis  Declares which values shall be plotted on the Y axis: If y.axis = "r", the Y axis shows the regional growth. If y.axis = "n", the Y axis shows the national growth. To set both growths in ratio, choose y.axis = "rn" (regional vs. national growth)
psize  Point size in the portfolio matrix plot (mostly the absolute values of employment in i industries in region j at time 2)
psize.factor  Enlargement factor for the points in the plot
time.periods  No. of regarded time periods (for average growth rates)
pxm  Name of the X axis in the plot
pmy  Name of the Y axis in the plot
ptitle  Plot title
The portfolio matrix is a graphic tool displaying the development of one variable compared to another variable. The plot shows the regarded variable on the $x$ axis and a variable with which it is confronted on the $y$ axis while the graph is divided in four quadrants. Originally, the portfolio matrix was developed by the Boston Consulting Group to analyze the performance of product lines in marketing, also known as the growth-share matrix. The quadrants show the performance of the regarded objects (stars, cash cows, question marks, dogs) (Henderson 1973). But the portfolio matrix can also be used to analyze/illustrate the world market integration of a region or a national economy by confronting e.g. the increase in world market share ($x$ axis) and the world trade growth ($y$ axis) (Baker et al. 2002). Another option is to analyze/illustrate the economic performance of a region (Howard 2007). E.g. it is possible to confront the growth of industries in a region with the all-over growth of these industries in the national economy.

This function is a special case of portfolio matrix, showing the regional specialization on the X axis instead of the regional growth (which can be plotted on the Y axis).

**Value**

A portfolio matrix plot.

Invisible: a list containing the following items:

- **portfolio.data**
  - The data related to the plot
- **locq**
  - The localization quotients for each year
- **growth**
  - The growth values for each industry

**Author(s)**

Thomas Wieland
locq2

References


See Also

locq, portfolio, shift, shiftd, shifti

data(Goettingen)
# Loads employment data for Goettingen and Germany (2008-2017)
locq.growth(Goettingen$Goettingen2008[2:16], Goettingen$Goettingen2017[2:16],
Goettingen$BRD2008[2:16], Goettingen$BRD2017[2:16],
psize = Goettingen$Goettingen2017[2:16],
industry.names = Goettingen$WA_WZ2008[2:16], pcol.border = "grey",
leg = TRUE, leg.fsize = 0.4, leg.x = -0.2)

locq2

Description

Calculating the location quotient (a.k.a. Hoover-Balassa quotient) for a given number of I industries and J regions

Usage

locq2(e_ij, industry.id, region.id, LQ.norm = "none",
LQ.output = "mat", na.rm = TRUE)

Arguments

a vector with the employment of industry i in region j
industry.id a vector containing the IDs of the industries i
region.id a vector containing the IDs of the regions j
Type of normalization of the location quotients: no normalization (default: LQ.norm = "none"), z values (LQ.norm = "OG") or z values of logged location quotients (LQ.norm = "T")
The location quotient is a simple measure for the concentration of an industry \((i)\) in a region \((j)\) and is also the mathematical basis for other related indicators in regional economics (e.g. \texttt{gini.conc()}). The function returns the value \(LQ\) which is equal to 1 if the concentration of the regarded industry is exactly the same as the over-all concentration (that means, it is proportionally represented in region \(j\)). If the value of \(LQ\) is smaller (bigger) than 1, the industry is underrepresented (overrepresented). The function checks the input values for errors (i.e. if employment in a region is bigger than over-all employment).

Two types of normalization are available: \(z\) values of the location quotients (O’Donoghue/Gleave 2004) or \(z\) values of logged location quotients (Tian 2013).

### Value

A matrix or data frame containing \(I \times J\) values of \(LQ\)

### Author(s)

Thomas Wieland

### References


### See Also

\texttt{litzenberger}, \texttt{gini.conc}, \texttt{gini.spec}, \texttt{locq}, \texttt{hoover}, \texttt{ellison.a}, \texttt{ellison.a2}, \texttt{ellison.c}, \texttt{ellison.c2}

### Examples

```r
data (G.regions.industries)

lqs <- locq2(e_ij = G.regions.industries$emp_all,
```

---
G.regions.industries$ind_code, G.regions.industries$region_code,
LQ.output = "df")
# output as data frame

lqs_sort <- lqs[order(lqs$LQ, decreasing = TRUE),]
# Sort decreasing by size of LQ

lqs_sort[1:5,]

---

### lorenz

**Lorenz curve**

**Description**

Calculating and plotting the Lorenz curve

**Usage**

```r
lorenz(x, weighting = NULL, z = NULL, na.rm = TRUE,
  lcx = "% of objects", lcy = "% of regarded variable",
  lctitle = "Lorenz curve", le.col = "blue", lc.col = "black",
  lsize = 1.5, ltype = "solid", bg.col = "gray95", bgrid = TRUE,
  bgrid.col = "white", bgrid.size = 2, bgrid.type = "solid",
  lcg = FALSE, lcg.n = FALSE, lcg.caption = NULL, lcg.lab.x = 0,
  lcg.lab.y = 1, add.lc = FALSE, plot.lc = TRUE)
```

**Arguments**

- `x` A numeric vector (e.g. dataset of household income, sales turnover or supply)
- `weighting` A numeric vector containing the weighting data (e.g. size of income classes when calculating a Lorenz curve for aggregated income data)
- `z` A numeric vector for (optionally) comparing the cumulative distribution
- `na.rm` logical argument that indicates whether NA values should be excluded before computing results
- `lcx` defines the x axis label
- `lcy` defines the y axis label
- `lctitle` defines the overall title of the Lorenz curve plot
- `le.col` defines the color of the diagonale (line of equality)
- `lc.col` defines the color of the Lorenz curve
- `lsize` defines the size of the lines (default: 1)
- `ltype` defines the type of the lines (default: "solid")
- `bg.col` defines the background color of the plot (default: "gray95")
- `bgrid` logical argument that indicates if a grid is shown in the plot
bgrid.col if bgrid = TRUE (background grid), bgrid.col defines the color of the background grid (default: "white")

bgrid.size if bgrid = TRUE (background grid), bgrid.size defines the size of the background grid (default: 2)

bgrid.type if bgrid = TRUE (background grid), bgrid.type defines the type of lines of the background grid (default: "solid")

lcg logical argument that indicates if the non-standardized Gini coefficient is displayed in the Lorenz curve plot

lcgn logical argument that indicates if the standardized Gini coefficient is displayed in the Lorenz curve plot

lcg.caption specifies the caption above the coefficients

lcg.lab.x specifies the x coordinate of the label

lcg.lab.y specifies the y coordinate of the label

add.lc specifies if a new Lorenz curve is plotted (add.lc = "FALSE") or the plot is added to an existing Lorenz curve plot (add.lc = "TRUE")

plot.lc logical argument that indicates if the Lorenz curve itself is plotted (if plot.lc = FALSE, only the line of equality is plotted)

Details

The Gini coefficient (Gini 1912) is a popular measure of statistical dispersion, especially used for analyzing inequality or concentration. The Lorenz curve (Lorenz 1905), though developed independently, can be regarded as a graphical representation of the degree of inequality/concentration calculated by the Gini coefficient (G) and can also be used for additional interpretations of it. In an economic-geographical context, these methods are frequently used to analyse the concentration/inequality of income or wealth within countries (Aoyama et al. 2011). Other areas of application are analyzing regional disparities (Lessmann 2005, Nakamura 2008) and concentration in markets (sales turnover of competing firms) which makes Gini and Lorenz part of economic statistics in general (Doersam 2004, Roberts 2014).

The Gini coefficient (G) varies between 0 (no inequality/concentration) and 1 (complete inequality/concentration). The Lorenz curve displays the deviations of the empirical distribution from a perfectly equal distribution as the difference between two graphs (the distribution curve and a diagonal line of perfect equality). This function calculates G and plots the Lorenz curve optionally. As there are several ways to calculate the Gini coefficient, this function uses the formula given in Doersam (2004). Because the maximum of G is not equal to 1, also a standardized coefficient (G+) with a maximum equal to 1 can be calculated alternatively. If a Lorenz curve for aggregated data (e.g. income classes with averaged incomes) or the Lorenz curve has to be weighted, use a weighting vector (e.g. size of the income classes).

Value

A plot of the Lorenz curve.

Author(s)

Thomas Wieland
References


See Also

cv, gini.conc, gini.spec, herf, hoover

Examples

# Market concentration (example from Doersam 2004):
sales <- c(20, 50, 20, 10)
# sales turnover of four car manufacturing companies
lorenz (sales, lcx = "percentage of companies", lcy = "percentage of sales",
lctitle = "Lorenz curve of sales", lcg = TRUE, lcgn = TRUE)
# plots the Lorenz curve with user-defined title and labels
# including Gini coefficient

# Income classes (example from Doersam 2004):
income <- c(500, 1500, 2500, 4000, 7500, 15000)
# average income of 6 income classes
sizeofclass <- c(1000, 1200, 1600, 400, 200, 600)
# size of income classes
lorenz (income, weighting = sizeofclass, lcg = TRUE, lcgn = TRUE)
# plots the Lorenz curve with user-defined title and labels
# including Gini coefficient

# Regional disparities in Germany:
gdp <- c(460.69, 549.19, 124.16, 65.29, 31.59, 109.27, 263.44, 39.87, 258.53,
645.59, 131.95, 35.83, 112.66, 56.22, 85.61, 56.81)
# GDP of german regions 2015 (in billion EUR)
lorenz (gdp, lcg = TRUE, lcgn = TRUE)
mean2

Description
Calculating the arithmetic mean, weighted or non-weighted, or the geometric mean

Usage
mean2(x, weighting = NULL, output = "mean", na.rm = TRUE)

Arguments
x a numeric vector
weighting a numeric vector containing weighting data to compute the weighted arithmetic mean (instead of the non-weighted)
output argument to specify the output (output = "mean" returns the arithmetic mean, output = "geom" returns the geometric mean)
na.rm logical argument that whether NA values should be extracted or not

Details
This function uses the formula for the weighted arithmetic mean from Sheret (1984).

Value
Single numeric value. If output = "mean" and weighting is specified, the function returns a weighted arithmetic mean. If output = "geom", the geometric mean is returned.

Author(s)
Thomas Wieland

References

See Also
sd2
mssd

Examples

```r
avector <- c(5, 17, 84, 55, 39)
mean(avector)
mean2(avector)

wvector <- c(9, 757, 44, 18, 682)
mean2 (avector, weighting = wvector)
mean2 (avector, output = "geom")
```

---

mssd  
*Mean square successive difference*

Description

Calculating the mean square successive difference

Usage

```r
mssd (x)
```

Arguments

- `x`  
a numeric vector arranged in chronological order

Details

The *mean square successive difference*, $\delta^2$, is a dimensionless measure of variability over time (von Neumann et al. 1941). It can be used for assessing the volatility of a variable with respect to different subjects/groups.

Value

Single numeric value (the *mean square successive difference*, $\delta^2$).

Author(s)

Thomas Wieland

References


See Also

`var2, sd2, cv`
Examples

data1 <- c(10,10,10,20,20,20,30,30,30)
# stable growth
data2 <- c(20,10,30,10,30,20,30,20,10)
# high variability

# Means:
mean2(data1)
mean2(data2)
# Same means

# Standard deviation:
sd2(data1)
sd2(data2)
# Coefficient of variation:
cv(data1)
cv(data2)
# Measures of statistical dispersion are equal

mssd(data1)
mssd(data2)
# high differences in variability

portfolio

**Portfolio matrix**

Description

Portfolio matrix plot comparing two numeric vectors

Usage

portfolio(e_ij1, e_ij2, e_i1, e_i2, industry.names = NULL, psize, psize.factor = 10, time.periods = NULL, pmx = "Regional growth", pmy = "National growth", pmtitle = "Portfolio matrix", pcol = NULL, pcol.border = NULL, leg = FALSE, leg.fsize = 1, leg.col = NULL, leg.x = -max_val, leg.y = -max_val*1.5, bg.col = "gray95", bgrid = TRUE, bgrid.col = "white", bgrid.size = 2, bgrid.type = "solid", seg.x = 0, seg.y = 0)

Arguments

e_ij1 a numeric vector with i values containing the employment in i industries in region j at time 1
e_ij2 a numeric vector with i values containing the employment in i industries in region j at time 2
portfolio

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e_i1</td>
<td>a numeric vector with ( i ) values containing the total employment in ( i ) industries at time 1</td>
</tr>
<tr>
<td>e_i2</td>
<td>a numeric vector with ( i ) values containing the total employment in ( i ) industries at time 2</td>
</tr>
<tr>
<td>industry.names</td>
<td>Industry names (e.g. from the relevant statistical classification of economic activities)</td>
</tr>
<tr>
<td>psize</td>
<td>Point size in the portfolio matrix plot (mostly the absolute values of employment in ( i ) industries in region ( j ) at time 2)</td>
</tr>
<tr>
<td>psize.factor</td>
<td>Enlargement factor for the points in the plot</td>
</tr>
<tr>
<td>time.periods</td>
<td>No. of regarded time periods (for average growth rates)</td>
</tr>
<tr>
<td>pmx</td>
<td>Name of the X axis in the plot</td>
</tr>
<tr>
<td>pmy</td>
<td>Name of the Y axis in the plot</td>
</tr>
<tr>
<td>pmtitle</td>
<td>Plot title</td>
</tr>
<tr>
<td>pcol</td>
<td>Industry-specific point colors</td>
</tr>
<tr>
<td>pcol.border</td>
<td>Color of point border</td>
</tr>
<tr>
<td>leg</td>
<td>Logical argument that indicates if a legend has to be added to the plot</td>
</tr>
<tr>
<td>leg.fsize</td>
<td>If leg = TRUE: Font size in the plot legend</td>
</tr>
<tr>
<td>leg.col</td>
<td>No. of columns in the legend</td>
</tr>
<tr>
<td>leg.x</td>
<td>If leg = TRUE: X coordinate of the legend</td>
</tr>
<tr>
<td>leg.y</td>
<td>If leg = TRUE: Y coordinate of the legend</td>
</tr>
<tr>
<td>bg.col</td>
<td>Background color</td>
</tr>
<tr>
<td>bgrid</td>
<td>Logical argument that indicates if a grid has to be added to the plot</td>
</tr>
<tr>
<td>bgrid.col</td>
<td>If bgrid = TRUE: Color of the grid</td>
</tr>
<tr>
<td>bgrid.size</td>
<td>If bgrid = TRUE: Size of the grid</td>
</tr>
<tr>
<td>bgrid.type</td>
<td>If bgrid = TRUE: Type of the grid</td>
</tr>
<tr>
<td>seg.x</td>
<td>X coordinate of segmentation of the plot</td>
</tr>
<tr>
<td>seg.y</td>
<td>Y coordinate of segmentation of the plot</td>
</tr>
</tbody>
</table>

Details

The portfolio matrix is a graphic tool displaying the development of one variable compared to another variable. The plot shows the regarded variable on the \( x \) axis and a variable with which it is confronted on the \( y \) axis while the graph is divided in four quadrants. Originally, the portfolio matrix was developed by the Boston Consulting Group to analyze the performance of product lines in marketing, also known as the growth-share matrix. The quadrants show the performance of the regarded objects (stars, cash cows, question marks, dogs) (Henderson 1973). But the portfolio matrix can also be used to analyze/illustrate the world market integration of a region or a national economy by confronting e.g. the increase in world market share (\( x \) axis) and the world trade growth (\( y \) axis) (Baker et al. 2002). Another option is to analyze/illustrate the economic performance of a region (Howard 2007). E.g. it is possible to confront the growth of industries in a region with the all-over growth of these industries in the national economy.
Value

A portfolio matrix plot and a data frame containing the related data (invisible).

Author(s)

Thomas Wieland

References


See Also

`shift`, `shiftf`, `shifti`

Examples

data(Freiburg)
# Loads employment data for Freiburg and Germany (2008 and 2014)

portfolio(Freiburg$e_Freiburg2008, Freiburg$e_Freiburg2014, Freiburg$e_Germany2008, Freiburg$e_Germany2014,
industry.names = Freiburg$industry, Freiburg$e_Freiburg2014, psize.factor = 12,
px = "Freiburg", pmy = "Deutschland", pmtitle = "Freiburg und BRD",
pcol = Freiburg$color, leg = TRUE, leg.fsize = 0.6, bgrid = TRUE, leg.y = -0.17)

rca

Analysis of regional beta and sigma convergence

Description

This function provides the analysis of absolute and conditional regional economic beta convergence and sigma convergence for cross-sectional data. Beta convergence can be estimated using an OLS or NLS technique. Sigma convergence can be analyzed using ANOVA or trend regression.
Usage

rca(gdp1, time1, gdp2, time2, 
  conditions = NULL, conditions.formula = NULL, conditions.startval = NULL, 
  beta.estimate = "ols", beta.plot = FALSE, beta.plotPSize = 1, beta.plotPCol = "black", 
  beta.plotLine = FALSE, beta.plotLineCol = "red", beta.plotX = "Ln (initial)", 
  beta.plotY = "Ln (growth)", beta.plotTitle = "Beta convergence", beta.bgCol = "gray95", 
  beta.bgrid = TRUE, beta.bgridCol = "white", beta.bgridSize = 2, beta.bgridType = "solid", 
  sigma.type = "anova", sigma.measure = "sd", sigma.log = TRUE, sigma.weighting = NULL, 
  sigma.issample = FALSE, sigma.plot = FALSE, sigma.plotLSize = 1, 
  sigma.plotLineCol = "black", sigma.plotRLine = FALSE, sigma.plotRLineCol = "blue", 
  sigma.Ymin = 0, sigma.plotX = "Time", sigma.plotY = "Variation", 
  sigma.plotTitle = "Sigma convergence", sigma.bgCol = "gray95", sigma.bgrid = TRUE, 
  sigma.bgridCol = "white", sigma.bgridSize = 2, sigma.bgridType = "solid")

Arguments

gdp1 A numeric vector containing the GDP per capita (or another economic variable) 
at time \( t \)
time1 A single value of time \( t (= \text{the initial year}) \)
gdp2 A numeric vector containing the GDP per capita (or another economic variable) 
at time \( t+1 \) or a data frame containing the GDPs per capita (or another economic variable) 
at time \( t+1, t+2, t+3, \ldots, t+n \)
time2 A single value of time \( t+1 \) or \( t_n, \) respectively
conditions A data frame containing the conditions for conditional beta convergence
conditions.formula If beta.estimate = "nls": A formula for the functional linkage of the conditions 
in the case of conditional beta convergence
conditions.startval If beta.estimate = "nls": Starting values for the parameters of the conditions 
in the case of conditional beta convergence
beta.estimate Beta estimate via ordinary least squares (OLS) or nonlinear least squares (NLS). 
Default: beta.estimate = "ols"
beta.plot Boolean argument that indicates if a plot of beta convergence has to be created
beta.plotPSize If beta.plot = TRUE: Point size in the beta convergence plot
beta.plotPCol If beta.plot = TRUE: Point color in the beta convergence plot
beta.plotLine If beta.plot = TRUE: Logical argument that indicates if a regression line has to 
be added to the plot
beta.plotLineCol If beta.plot = TRUE and beta.plotLine = TRUE: Line color of regression line
beta.plotX If beta.plot = TRUE: Name of the X axis
beta.plotY If beta.plot = TRUE: Name of the Y axis
beta.plotTitle If beta.plot = TRUE: Plot title
beta.bgCol If beta.plot = TRUE: Plot background color
beta.bgrid If \texttt{beta.plot = TRUE}: Logical argument that indicates if the plot contains a grid
beta.bgridCol If \texttt{beta.plot = TRUE} and \texttt{beta.bgrid = TRUE}: Color of the grid
beta.bgridSize If \texttt{beta.plot = TRUE} and \texttt{beta.bgrid = TRUE}: Size of the grid
beta.bgridType If \texttt{beta.plot = TRUE} and \texttt{beta.bgrid = TRUE}: Type of the grid

\texttt{sigma.type} Estimating sigma convergence via ANOVA (two years) or trend regression (more than two years). Default: \texttt{sigma.type = "anova"}

\texttt{sigma.measure} argument that indicates how the sigma convergence should be measured. The default is \texttt{output = "sd"}, which means that the standard deviation is used. If \texttt{output = "var"} or \texttt{output = "cv"}, the variance or the coefficient of variation is used, respectively.

\texttt{sigma.log} Logical argument. Per default (\texttt{sigma.log = TRUE}), also in the sigma convergence analysis, the economic variables are transformed by natural logarithm. If the original values should be used, state \texttt{sigma.log = FALSE}

\texttt{sigma.weighting} If the measure of statistical dispersion in the sigma convergence analysis (coefficient of variation or standard deviation) should be weighted, a weighting vector has to be stated

\texttt{sigma.issample} Logical argument that indicates if the dataset is a sample or the population (default: \texttt{is.sample = FALSE}, so the denominator of variance is \( n \))

\texttt{sigma.plot} Logical argument that indicates if a plot of sigma convergence has to be created
\texttt{sigma.plotLSIZE} If \texttt{sigma.plot = TRUE}: Line size of the sigma convergence plot
\texttt{sigma.plotLineColor} If \texttt{sigma.plot = TRUE}: Line color of the sigma convergence plot
\texttt{sigma.plotRLine} If \texttt{sigma.plot = TRUE}: Logical argument that indicates if a regression line has to be added to the plot
\texttt{sigma.plotRLineColor} If \texttt{sigma.plot = TRUE} and \texttt{sigma.plotRLine = TRUE}: Color of the regression line
\texttt{sigma.Ymin} If \texttt{sigma.plot = TRUE}: start value of the Y axis in the plot
\texttt{sigma.plotX} If \texttt{sigma.plot = TRUE}: Name of the X axis
\texttt{sigma.plotY} If \texttt{sigma.plot = TRUE}: Name of the Y axis
\texttt{sigma.plotTitle} If \texttt{sigma.plot = TRUE}: Title of the plot

\texttt{sigma.bgCol} If \texttt{sigma.plot = TRUE}: Plot background color
\texttt{sigma.bgrid} If \texttt{sigma.plot = TRUE}: Logical argument that indicates if the plot contains a grid
\texttt{sigma.bgridCol} If \texttt{sigma.plot = TRUE} and \texttt{sigma.bgrid = TRUE}: Color of the grid
\texttt{sigma.bgridSize} If \texttt{sigma.plot = TRUE} and \texttt{sigma.bgrid = TRUE}: Size of the grid
\texttt{sigma.bgridType} If \texttt{sigma.plot = TRUE} and \texttt{sigma.bgrid = TRUE}: Type of the grid
Details

From the regional economic perspective (in particular the neoclassical growth theory), regional disparities are expected to decline. This convergence can have different meanings: Sigma convergence ($\sigma$) means a harmonization of regional economic output or income over time, while beta convergence ($\beta$) means a decline of dispersion because poor regions have a stronger economic growth than rich regions (Capello/Nijkamp 2009). Regardless of the theoretical assumptions of a harmonization in reality, the related analytical framework allows to analyze both types of convergence for cross-sectional data (GDP p.c. or another economic variable, $y$, for $i$ regions and two points in time, $t$ and $t + T$), or one starting point ($t$) and the average growth within the following $n$ years ($t + 1, t + 2, ..., t + n$), respectively. Beta convergence can be calculated either in a linearized OLS regression model or in a nonlinear regression model. When no other variables are integrated in this model, it is called absolute beta convergence. Implementing other region-related variables (conditions) into the model leads to conditional beta convergence. If there is beta convergence ($\beta < 0$), it is possible to calculate the speed of convergence, $\lambda$, and the so-called Half-Life $H$, while the latter is the time taken to reduce the disparities by one half (Allington/McCombie 2007, Goecke/Huether 2016). There is sigma convergence, when the dispersion of the variable ($\sigma$), e.g. calculated as standard deviation or coefficient of variation, reduces from $t$ to $t + T$. This can be measured using ANOVA for two years or trend regression with respect to several years (Furceri 2005, Goecke/Huether 2016).

The rca function is a wrapper for the functions betaconv.ols, betaconv.nls, sigmaconv and sigmaconv.t. This function calculates (absolute and/or conditional) beta convergence and sigma convergence. Regional disparities are measured by the standard deviation (or variance, coefficient of variation) for all GDPs per capita (or another economic variable) for the given years. Beta convergence is estimated either using ordinary least squares (OLS) or nonlinear least squares (NLS). If the beta coefficient is negative (using OLS) or positive (using NLS), there is beta convergence. Sigma convergence is analyzed either using an analysis of variance (ANOVA) for these deviation measures (year 1 divided by year 2, F-statistic) or a trend regression (F-statistic, t-statistic). In the former case, if $\sigma_1/\sigma_2 > 0$, there is sigma convergence. In the latter case, if the slope of the trend regression is negative, there is sigma convergence.

Value

A list containing the following objects:

- betaconv: A list containing the following objects:
  - regdata: A data frame containing the regression data, including the ln-transformed economic variables
  - tinterval: The time interval
  - abeta: A list containing the estimates of the absolute beta convergence regression model, including lambda and half-life
  - cbeta: If conditions are stated: a list containing the estimates of the conditional beta convergence regression model, including lambda and half-life
- sigmaconv: A list containing the following objects:
  - sigmaconv: A matrix containing either the standard deviations, their quotient and the results of the significance test (F-statistic) or the results of trend regression
Author(s)

Thomas Wieland

References


See Also

betaconv.ols, betaconv.nls, betaconv.speed, sigmaconv, sigmaconv.t, cv, sd2, var2

Examples

data(G.counties.gdp)
# Loading GDP data for Germany (counties = Landkreise)

# Two years, no conditions (Absolute beta convergence)

regionaldummies <- to.dummy(G.counties.gdp$regional)
# Creating dummy variables for West/East
G.counties.gdp$West <- regionaldummies[,2]
G.counties.gdp$East <- regionaldummies[,1]
# Adding dummy variables to data

# Two years, with conditions
# (Absolute and conditional beta convergence)

# Store results in object
converg1$betaconv$abeta
# Addressing estimates for the conditional beta model

rca (G.counties.gdp$gdppc2010, 2010, G.counties.gdp[65:68], 2014, conditions = NULL,
sigma.type = "trend", beta.plot = TRUE, sigma.plot = TRUE)
# Five years, no conditions (Absolute beta convergence)
# with plots for both beta and sigma convergence

---

**reilly**  
*Law of retail gravitation by Reilly*

**Description**

Calculating the proportion of sales from an intermediate town between two cities or retail locations

**Usage**

```
reilly(P_a, P_b, D_a, D_b, gamma = 1, lambda = 2, relation = NULL)
```

**Arguments**

- **P_a**: a single numeric value of attractivity/population size of location/city *a*
- **P_b**: a single numeric value of attractivity/population size of location/city *b*
- **D_a**: a single numeric value of the distance from the intermediate town to location/city *a*
- **D_b**: a single numeric value of the distance from the intermediate town to location/city *b*
- **gamma**: a single numeric value for the exponential weighting of size (default: 1)
- **lambda**: a single numeric value for the exponential weighting of distance (transport costs, default: -2)
- **relation**: a single numeric value containing the relation of trade between cities/locations *a* and *b* (only needed if the distance decay parameters has to be estimated instead of the sales flows)

**Details**

The *law of retail gravitation* by Reilly (1929, 1931) was the first *spatial interaction model* for retailing and services. This "law" states that two cities/locations attract customers from an intermediate town proportionally to the attractivity/population size of the two cities/locations and in inverse proportion to the squares of the transport costs (e.g. distance, travelling time) from these two locations to the intermediate town. But both variables can be weighted by exponents. The distance exponent can also be derived from empirical data (if an empirical relation is stated). The *breaking point formula* by Converse (1949) is a separate transformation of Reilly’s law (see the function converse). The models by Reilly and Converse are simple *spatial interaction models* and are considered as
deterministic market area models due to their exact allocation of demand origins to locations. A probabilistic approach including a theoretical framework was developed by Huff (1962) (see the function huff).

Value

If no relation is stated, a list with three values:

- relation_AB: relation of trade between cities/locations a and b
- prop_A: proportion of city/location a
- prop_B: proportion of city/location b

If a relation is stated instead of weighting parameters, a single numeric value containing the estimated distance decay parameter.

Author(s)

Thomas Wieland

References


See Also

huff, converse

Examples

# Example from Converse (1949):
reilly (39851, 37366, 27, 25)
# two cities (pop. size 39.851 and 37.366)
# with distances of 27 and 25 miles to intermediate town
myresults <- reilly (39851, 37366, 27, 25)
myresults$prop_A
# proportion of location a
# Distance decay parameter for the given sales relation:
reilly (39851, 37366, 27, 25, gamma = 1, lambda = NULL, relation = 0.9143555)
# returns 2

**ripley**

**Ripley’s K**

**Description**

Analyzing point clustering with Ripley’s K function

**Usage**

```r
ripley(loc_df, loc_id, loc_lat, loc_lon,
area, t.max, t.sep = 10, K.local = FALSE,
ci.boot = FALSE, ci.alpha = 0.05, ciboot.samples = 100,
progmsg = FALSE, K.plot = TRUE, Kplot.func = "K",
plot.title = "Ripley’s K", plotX = "t",
plotY = paste(Kplot.func, "Observed vs. expected"),
lcol.exp = "blue", lcol.emp = "red", lsize.exp = 1,
ltype.exp = "solid", lsize.emp = 1, ltype.emp = "solid",
bg.col = "gray95", bgrid = TRUE, bgrid.col = "white",
bgrid.size = 2, bgrid.type = "solid")
```

**Arguments**

- **loc_df** A data frame containing the points
- **loc_id** Column containing the IDs of the points in the data frame loc_df
- **loc_lat** Column containing the latitudes of the points in the data frame loc_df
- **loc_lon** Column containing the longitudes of the points in the data frame loc_df
- **area** Total area of the regarded region
- **t.max** Maximum distance
- **t.sep** Number of distance intervals
- **K.local** Logical arguments that indicates whether local K values are computed or not
- **ci.boot** Logical arguments that indicates whether bootstrap confidence intervals are computed or not
- **ci.alpha** Significance level of the bootstrap confidence intervals
- **ciboot.samples** No. of bootstrap samples
- **progmsg** Logical argument: Printing progress messages or not
- **K.plot** Logical argument: Plot K function or not
- **Kplot.func** Which function has to be plotted? K function (Kplot.func = "K"), L function (Kplot.func = "L") or H function (Kplot.func = "H")
- **plot.title** If K.plot = TRUE: Plot title
plotX  If K.plot = TRUE: name of the X axis
plotY  If K.plot = TRUE: name of the Y axis
lcol.exp If K.plot = TRUE: color of the line representing the expected values
lcol.emp If K.plot = TRUE: color of the line representing the empirical values
lsize.exp If K.plot = TRUE: size of the line representing the expected values
lsize.emp If K.plot = TRUE: size of the line representing the empirical values
ltype.exp If K.plot = TRUE: type of the line representing the expected values
ltype.emp If K.plot = TRUE: type of the line representing the empirical values
bg.col  if lc = TRUE (plot of Lorenz curve), bg.col defines the background color of the plot (default: "gray95")
bgrid  if lc = TRUE (plot of Lorenz curve), the logical argument bgrid defines if a grid is shown in the plot
bgrid.col if lc = TRUE (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.col defines the color of the background grid (default: "white")
bgrid.size if lc = TRUE (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.size defines the size of the background grid (default: 2)
bgrid.type if lc = TRUE (plot of Lorenz curve) and bgrid = TRUE (background grid), bgrid.type defines the type of lines of the background grid (default: "solid")

Details
Calculating and plotting of the K function and its derivations (L function, H function) and, optionally, bootstrap confidence intervals.

Value
The function returns a list containing:

K  A data.frame containing the K/L/H/t values
K_local  A data.frame containing the local K values (if stated)
local_ci  A data.frame containing the local confidence intervals (if stated)

Author(s)
Thomas Wieland

References

See Also
dist, dist.buf, dist.mat
Examples

## Not run:
data(GoettingenHealth1)
# general practitioners, psychotherapists and pharmacies
area_goe <- 1753000000
# area of Landkreis Goettingen (sqm)
area_nom <- 1267000000
# area of Landkreis Northeim (sqm)
area_gn <- area_goe+area_nom
sqrt(area_gn/pi)

# this takes some seconds
ripley(GoettingenHealth1[GoettingenHealth1$type == "phys_gen",]
, "location", "lat", "lon", area = area_gn, t.max = 30000, t.sep = 300)
ripley(GoettingenHealth1[GoettingenHealth1$type == "pharm",]
, "location", "lat", "lon", area = area_gn, t.max = 30000, t.sep = 300)
ripley(GoettingenHealth1[GoettingenHealth1$type == "psych",]
, "location", "lat", "lon", area = area_gn, t.max = 30000, t.sep = 300)

## End(Not run)

sd2

---

**sd2**

*Standard deviation (extended)*

**Description**

Calculating the standard deviation (sd), weighted or non-weighted, for samples or populations

**Usage**

sd2 (x, is.sample = TRUE, weighting = NULL, wmean = FALSE, na.rm = TRUE)

**Arguments**

- **x**: a numeric vector
- **is.sample**: logical argument that indicates if the dataset is a sample or the population (default: is.sample = TRUE, so the denominator of variance is n − 1)
- **weighting**: a numeric vector containing weighting data to compute the weighted standard deviation (instead of the non-weighted sd)
- **wmean**: logical argument that indicates if the weighted mean is used when calculating the weighted standard deviation
- **na.rm**: logical argument that whether NA values should be extracted or not
Details

The function calculates the *standard deviation*. Unlike the R base `sd` function, the `sd2` function allows to choose if the data is treated as sample (denominator of variance is \( n - 1 \)) or not (denominator of variance is \( n \))

From a regional economic perspective, the sd is closely linked to the concept of *sigma convergence* (\( \sigma \)) which means a harmonization of regional economic output or income over time, while the other type of convergence, *beta convergence* (\( \beta \)), means a decline of dispersion because poor regions have a stronger growth than rich regions (Capello/Nijkamp 2009). The sd allows to summarize regional disparities (e.g. disparities in regional GDP per capita) in one indicator. The coefficient of variation (see the function cv) is more frequently used for this purpose (e.g. Lessmann 2005, Huang/Leung 2009, Siljak 2015). But the sd can also be used for any other types of disparities or dispersion, such as disparities in supply (e.g. density of physicians or grocery stores).

The standard deviation can be weighted by using a second weighting vector. As there is more than one way to weight measures of statistical dispersion, this function uses the formula for the weighted sd (\( \sigma_w \)) from Sheret (1984). The vector \( x \) is automatically treated as a sample (such as in the base sd function), so the denominator of variance is \( n - 1 \), if it is not, set `is.sample = FALSE`.

Value

Single numeric value. If weighting is specified, the function returns a weighted standard deviation (optionally using a weighted arithmetic mean if `wmean = TRUE`).

Author(s)

Thomas Wieland

References


Huang, Y./Leung, Y. (2009): “Measuring Regional Inequality: A Comparison of Coefficient of Variation and Hoover Concentration Index”. In: *The Open Geography Journal, 2*, p. 25-34.


See Also

`gini, herf, hoover, mean2, rca`
Examples

# Regional disparities / sigma convergence in Germany
data(G.counties.gdp)
# GDP per capita for German counties (Landkreise)
sd_gdppc <- apply(G.counties.gdp[54:68], MARGIN = 2, FUN = sd2)
# Calculating standard deviation for the years 2000-2014
years <- 2000:2014
# vector of years (2000-2014)
plot(years, sd_gdppc, "l", ylim = c(0,15000), xlab = "Year",
ylab = "SD of GDP per capita")
# Plot sd over time

### Shift-share analysis

Description

Analyzing regional growth with the shift-share analysis

Usage

shift(e_ij1, e_ij2, e_i1, e_i2, industry.names = NULL,
shift.method = "Dunn", print.results = TRUE, plot.results = FALSE,
plot.colours = NULL, plot.title = NULL, plot.portfolio = FALSE, ...)

Arguments

e_ij1  a numeric vector with i values containing the employment in i industries in
region j at time 1

e_ij2  a numeric vector with i values containing the employment in i industries in
region j at time 2

e_i1  a numeric vector with i values containing the total employment in i industries at
     time 1

e_i2  a numeric vector with i values containing the total employment in i industries at
     time 2

industry.names  Industry names (e.g. from the relevant statistical classification of economic ac-
     tivities)

shift.method  Method of shift-share-analysis to be used ("Dunn", "Esteban", "Gerfin") (de-
     fault: shift.method = "Dunn")

print.results  Logical argument that indicates if the function shows the results or not

plot.results  Logical argument that indicates if the results have to be plotted

plot.colours  If plot.results = TRUE: Plot colours

plot.title  If plot.results = TRUE: Plot title

plot.portfolio  Logical argument that indicates if the results have to be plotted in a portfolio
     matrix additionally

...  Additional arguments for the portfolio plot (see the function portfolio)
Details

The shift-share analysis (Dunn 1960) addresses the regional growth (or decline) regarding the overall development in the national economy. The aim of this analysis model is to identify which parts of the regional economic development can be traced back to national trends, effects of the regional industry structure and (positive) regional factors. The growth (or decline) of regional employment consists of three factors: \( l_{t+1} - l_t = nps + nds + nts \), where \( l_t \) is the employment in the region at time \( t \) and \( t + 1 \), respectively, and \( nps \) is the net proportionality shift, \( nds \) is the net differential shift and \( nts \) is the net total shift. Other variants are e.g. the shift-share method by Gerfin (Index method), the dynamic shift-share analysis (Barff/Knight 1988) or the extension by Esteban-Marquillas (1972).

As there is more than one way to calculate a Dunn-type shift-share analysis and the terms are not used consequently in the regional economic literature, this function and the documentation use the formulae and terms given in Farhauer/Kroell (2013). If \( \text{shift.method} = \text{"Dunn"} \), this function calculates the net proportionality shift (\( nps \)), the net differential shift (\( nds \)) and the net total shift (\( nts \)) where the last one represents the residuum of (positive) regional factors.

This function calculates a shift-share analysis for two years.

Value

A list containing the following objects:

- \( \text{components} \) A matrix containing the shift-share components related to the chosen method
- \( \text{growth} \) A matrix containing the industry-specific growth values
- \( \text{method} \) The chosen method, e.g. "Dunn"

Author(s)

Thomas Wieland

References

shift.growth

See Also

portfolio, shiftd, shifti, shift.growth

Examples

# Example from Farhauer/Kroell (2013):
region_A_t <- c(90,20,10,60)
region_A_t1 <- c(100,40,10,55)
# data for region A (time t and t+1)
nation_X_t <- c(400,150,150,400)
nation_X_t1 <- c(440,210,135,480)
# data for the national economy (time t and t+1)
resultsA <- shift(region_A_t, region_A_t1, nation_X_t, nation_X_t1)
# results for region A
region_B_t <- c(60,30,30,40)
region_B_t1 <- c(85,55,40,35)
# data for region B (time t and t+1)
resultsB <- shift(region_B_t, region_B_t1, nation_X_t, nation_X_t1)
# results for region B
region_C_t <- c(250,100,110,300)
region_C_t1 <- c(255,115,85,390)
# data for region C (time t and t+1)
resultsC <- shift(region_C_t, region_C_t1, nation_X_t, nation_X_t1)
# results for region C

# Example Freiburg dataset
data(Freiburg)
# Loads the data
shift(Freiburg$e_Freiburg2008, Freiburg$e_Freiburg2014, Freiburg$e_Germany2008,
Freiburg$e_Germany2014)
# results for Freiburg and Germany (2008 vs. 2014)

shift.growth

Growth rates for shift-share analysis

Description

This function calculates industry-specific growth rates which are part of the shift-share analysis

Usage

shift.growth(e_ij1, e_ij2, e_i1, e_i2, time.periods = NULL,
industry.names = NULL)

Arguments

e_ij1 a numeric vector with i values containing the employment in i industries in
region j at time 1
Details

The shift-share analysis (Dunn 1960) addresses the regional growth (or decline) regarding the overall development in the national economy. The aim of this analysis model is to identify which parts of the regional economic development can be traced back to national trends, effects of the regional industry structure and (positive) regional factors. The growth (or decline) of regional employment consists of three factors: \( l_{t+1} - l_t = nps + nds + nts \), where \( l \) is the employment in the region at time \( t \) and \( t + 1 \), respectively, and \( nps \) is the net proportionality shift, \( nds \) is the net differential shift and \( nts \) is the net total shift. Other variants are e.g. the shift-share method by Gerfin (Index method) and the dynamic shift-share analysis (Barff/Knight 1988).

As there is more than one way to calculate a Dunn-type shift-share analysis and the terms are not used consequently in the regional economic literature, this function and the documentation use the formulae and terms given in Farhauer/Kroell (2013). If \( shift\_method = "Dunn" \), this function calculates the net proportionality shift (\( nps \)), the net differential shift (\( nds \)) and the net total shift (\( nts \)) where the last one represents the residuum of (positive) regional factors.

This function calculates industry-specific growth rates which are part of a shift-share analysis.

Value

A matrix containing the industry-specific growth values

Author(s)

Thomas Wieland

References


See Also
portfolio, shift, shiftd, shifti

Examples

# Example from Farhauer/Kroell (2013):
region_A_t <- c(90,20,10,60)
region_A_t1 <- c(100,40,10,55)
# data for region A (time t and t+1)
nation_X_t <- c(400,150,150,400)
nation_X_t1 <- c(440,210,135,480)
# data for the national economy (time t and t+1)
shift.growth(region_A_t, region_A_t1, nation_X_t, nation_X_t1)

shiftd

Dynamic shift-share analysis

Description

Analyzing regional growth with the dynamic shift-share analysis

Usage

shiftd(e_ij1, e_ij2, e_i1, e_i2, time1, time2,
industry.names = NULL, shift.method = "Dunn",
gerfin.shifts = "mean", print.results = TRUE,
plot.results = FALSE, plot.colours = NULL, plot.title = NULL,
plot.portfolio = FALSE, ...)

Arguments

e_ij1 a numeric vector with i values containing the employment in i industries in region j at time 1

e_ij2 a numeric data frame or matrix with i rows containing the employment in i industries in region j and t columns, representing t (t > 1) years
e_i1 a numeric vector with \(i\) values containing the total employment in \(i\) industries at time 1

e_i2 a numeric data frame or matrix with \(i\) rows containing the total employment in \(i\) industries and \(t\) columns, representing \(t\) \((t > 1)\) years

time1 Initial year
time2 Final year

industry.names Industry names (e.g. from the relevant statistical classification of economic activities)

shift.method Method of shift-share-analysis to be used ("Dunn", "Gerfin") (default: shift.method = "Dunn")
gerfin.shifts If shift.method = "Gerfin": Logical argument that indicates if the shifts are calculated as sums or as means (default: gerfin = "mean")

print.results Logical argument that indicates if the function shows the results or not

plot.results Logical argument that indicates if the results have to be plotted

plot.colours If plot.results = TRUE: Plot colours

plot.title If plot.results = TRUE: Plot title

plot.portfolio Logical argument that indicates if the results have to be plotted in a portfolio matrix additionally

... Additional arguments for the portfolio plot (see the function portfolio)

Details

The shift-share analysis (Dunn 1960) addresses the regional growth (or decline) regarding the overall development in the national economy. The aim of this analysis model is to identify which parts of the regional economic development can be traced back to national trends, effects of the regional industry structure and (positive) regional factors. The growth (or decline) of regional employment consists of three factors: \(l_{t+1} - l_t = nps + nds + nts\), where \(l\) is the employment in the region at time \(t\) and \(t + 1\), respectively, and \(nps\) is the net proportionality shift, \(nds\) is the net differential shift and \(nts\) is the net total shift. Other variants are e.g. the shift-share method by Gerfin (Index method) and the dynamic shift-share analysis (Barff/Knight 1988).

As there is more than one way to calculate a Dunn-type shift-share analysis and the terms are not used consequently in the regional economic literature, this function and the documentation use the formulae and terms given in Farhauer/Kroell (2013). If shift.method = "Dunn", this function calculates the net proportionality shift (nps), the net differential shift (nds) and the net total shift (nts) where the last one represents the residuum of (positive) regional factors.

This function calculates a dynamic shift-share analysis for at least two years.

Value

A list containing the following objects:

components A matrix containing the shift-share components related to the chosen method

components.year A matrix containing the shift-share components for each year

growth A matrix containing the industry-specific growth values

method The chosen method, e.g. "Dunn"
Author(s)
Thomas Wieland

References

See Also
portfolio, shift, shifti, shift.growth

Examples
# Example from Farhauer/Kroell (2013), extended:
region_A_t <- c(90,20,10,60)
region_A_t1 <- c(100,40,10,55)
region_A_t2 <- c(105,45,15,60)
# data for region A (time t and t+1)
nation_X_t <- c(400,150,150,400)
nation_X_t1 <- c(440,210,135,480)
nation_X_t2 <- c(460,230,155,500)
# data for the national economy (time t and t+1)
shiftd(region_A_t, data.frame(region_A_t1, region_A_t2), nation_X_t, data.frame(nation_X_t1, nation_X_t2), time1 = 2000, time2 = 2002, plot.results = TRUE, plot.portfolio = TRUE, psize = region_A_t1)

data(Goettingen)
shifti

Shift-share analysis for industries

Description

Analyzing industry-specific regional growth with the shift-share analysis

Usage

shifti(e_ij1, e_ij2, e_i1, e_i2, industry.names = NULL, shift.method = "Dunn", print.results = TRUE, plot.results = FALSE, plot.colours = NULL, plot.title = NULL, plot.portfolio = FALSE, ...)  

Arguments

e_ij1 a numeric vector with i values containing the employment in i industries in region j at time 1

e_ij2 a numeric vector with i values containing the employment in i industries in region j at time 2

e_i1 a numeric vector with i values containing the total employment in i industries at time 1

e_i2 a numeric vector with i values containing the total employment in i industries at time 2

industry.names Industry names (e.g. from the relevant statistical classification of economic activities)

shift.method Method of shift-share-analysis to be used ("Dunn", "Gerfin") (default: shift.method = "Dunn")

print.results Logical argument that indicates if the function shows the results or not

plot.results Logical argument that indicates if the results have to be plotted

plot.colours If plot.results = TRUE: Plot colours

plot.title If plot.results = TRUE: Plot title

plot.portfolio Logical argument that indicates if the results have to be plotted in a portfolio matrix additionally

... Additional arguments for the portfolio plot (see the function portfolio)

Details

The shift-share analysis (Dunn 1960) adresses the regional growth (or decline) regarding the overall development in the national economy. The aim of this analysis model is to identify which parts of the regional economic development can be traced back to national trends, effects of the regional industry structure and (positive) regional factors. The growth (or decline) of regional employment consists of three factors: \( l_{t+1} - l_t = nps + nds + nts \), where \( l \) is the employment in the region at time \( t \) and \( t + 1 \), respectively, and \( nps \) is the net proportionality shift, \( nds \) is the net differential
shift and nts is the net total shift. Other variants are e.g. the shift-share method by Gerfin (Index method) and the dynamic shift-share analysis (Barff/Knight 1988).

As there is more than one way to calculate a Dunn-type shift-share analysis and the terms are not used consequently in the regional economic literature, this function and the documentation use the formulae and terms given in Farhauer/Kroell (2013). If shift.method = "Dunn", this function calculates the net proportionality shift (nps), the net differential shift (nds) and the net total shift (nts) where the last one represents the residuum of (positive) regional factors.

This function calculates a shift-share analysis for at least two years and results industry-specific shift-share components.

Value
A list containing the following objects:

- components A matrix containing the shift-share components related to the chosen method
- components.industry A matrix containing the shift-share components for each industry
- growth A matrix containing the industry-specific growth values
- method The chosen method, e.g. "Dunn"

Author(s)
Thomas Wieland

References

See Also
portfolio, shift, shifti, shift.growth
Examples

# Example from Farhauer/Kroell (2013):
region_A_t <- c(90,20,10,60)
region_A_t1 <- c(100,40,10,55)
# data for region A (time t and t+1)
nation_X_t <- c(400,150,150,400)
nation_X_t1 <- c(440,210,135,480)
# data for the national economy (time t and t+1)
shiftid(region_A_t, region_A_t1, nation_X_t, nation_X_t1,
plot.results = TRUE, plot.portfolio = TRUE, psize = region_A_t1)

shiftid

Dynamic shift-share analysis for industries

Description

Analyzing industry-specific regional growth with the dynamic shift-share analysis

Usage

shiftid(e_ij1, e_ij2, e_i1, e_i2, time1, time2,
industry.names = NULL, shift.method = "Dunn",
gerfin.shifts = "mean", print.results = TRUE,
plot.results = FALSE, plot.colours = NULL, plot.title = NULL,
plot.portfolio = FALSE, ...)

Arguments

e_ij1  a numeric vector with i values containing the employment in i industries in
region j at time 1

e_ij2  a numeric data frame or matrix with i rows containing the employment in i
industries in region j and t columns, representing t (t > 1) years

e_i1   a numeric vector with i values containing the total employment in i industries at
time 1

e_i2   a numeric data frame or matrix with i rows containing the total employment in
i industries and t columns, representing t (t > 1) years

time1  Initial year

time2  Final year

industry.names Industry names (e.g. from the relevant statistical classification of economic ac-
tivities)

shift.method Method of shift-share-analysis to be used ("Dunn", "Gerfin") (default: shift.method = "Dunn")
gerfin.shifts If shift.method = "Gerfin": Logical argument that indicates if the shifts are
calculated as sums or as means (default: gerfin = "mean")

print.results Logical argument that indicates if the function shows the results or not
The `shift-share analysis` (Dunn 1960) addresses the regional growth (or decline) regarding the overall development in the national economy. The aim of this analysis model is to identify which parts of the regional economic development can be traced back to national trends, effects of the regional industry structure and (positive) regional factors. The growth (or decline) of regional employment consists of three factors: \( l_{t+1} - l_t = nps + nds + nts \), where \( l \) is the employment in the region at time \( t \) and \( t + 1 \), respectively, and \( nps \) is the net proportionality shift, \( nds \) is the net differential shift and \( nts \) is the net total shift. Other variants are e.g. the shift-share method by Gerfin (Index method) and the dynamic shift-share analysis (Barff/Knight 1988).

As there is more than one way to calculate a Dunn-type `shift-share analysis` and the terms are not used consequently in the regional economic literature, this function and the documentation use the formulae and terms given in Farhauer/Kroell (2013). If `shift.method = "Dunn"`, this function calculates the net proportionality shift (\( nps \)), the net differential shift (\( nds \)) and the net total shift (\( nts \)) where the last one represents the residuum of (positive) regional factors.

This function calculates a dynamic shift-share analysis for at least two years.

**Value**

A list containing the following objects:

- `components`: A matrix containing the shift-share components related to the chosen method
- `components.year`: A matrix containing the shift-share components for each year
- `growth`: A matrix containing the industry-specific growth values
- `method`: The chosen method, e.g. "Dunn"

**Author(s)**

Thomas Wieland

**References**


See Also

portfolio, shift, shifti, shift.growth

Examples

```r
# Example from Farhauer/Kroell (2013), extended:
region_A_t <- c(90,20,10,60)
region_A_t1 <- c(100,40,10,55)
region_A_t2 <- c(105,45,15,60)
# data for region A (time t and t+1)
nation_X_t <- c(400,150,150,400)
nation_X_t1 <- c(440,210,135,480)
nation_X_t2 <- c(460,230,155,500)
# data for the national economy (time t and t+1)
shiftd(region_A_t, data.frame(region_A_t1, region_A_t2), nation_X_t, 
data.frame(nation_X_t1, nation_X_t2), time1 = 2000, time2 = 2002, 
plot.results = TRUE, plot.portfolio = TRUE, psize = region_A_t1)

data(Goettingen)
shiftid(Goettingen$Goettingen2008[2:16], Goettingen[2:16,3:11], 
time1 = 2008, time2 = 2017, industry.names = Goettingen$WA_WZ2008[2:16], 
shift.method = "Dunn")
```

---

**Description**

Forecasting regional employment growth with the shift-share analysis (Gerfin model)
Usage

```r
shiftp(e_ij1, e_ij2, e_i1, e_i2, e_i3, time1, time2, time3,
industry.names = NULL, print.results = TRUE,
plot.results = FALSE, plot.colours = NULL, plot.title = NULL,
plot.portfolio = FALSE, ...)```

Arguments

- `e_ij1`: a numeric vector with `i` values containing the employment in `i` industries in region `j` at time 1
- `e_ij2`: a numeric vector with `i` values containing the employment in `i` industries in region `j` at time 2
- `e_i1`: a numeric vector with `i` values containing the total employment in `i` industries at time 1
- `e_i2`: a numeric vector with `i` values containing the total employment in `i` industries at time 2
- `e_i3`: a numeric vector with `i` values containing the total employment in `i` industries at time 3 (forecast value for total employment
- `time1`: start year (single value)
- `time2`: end year of empirical employment data (single value)
- `time3`: year of prognosis (single value)
- `industry.names`: Industry names (e.g. from the relevant statistical classification of economic activities)
- `print.results`: Logical argument that indicates if the function shows the results or not
- `plot.results`: Logical argument that indicates if the results have to be plotted
- `plot.colours`: If `plot.results = TRUE`: Plot colours
- `plot.title`: If `plot.results = TRUE`: Plot title
- `plot.portfolio`: Logical argument that indicates if the results have to be plotted in a portfolio matrix additionally
- `...`: Additional arguments for the portfolio plot (see the function `portfolio`)

Details

The shift-share analysis (Dunn 1960) addresses the regional growth (or decline) regarding the overall development in the national economy. The aim of this analysis model is to identify which parts of the regional economic development can be traced back to national trends, effects of the regional industry structure and (positive) regional factors. The growth (or decline) of regional employment consists of three factors: \( l_{t+1} - l_t = nps + nds + nts \), where \( l \) is the employment in the region at time \( t \) and \( t + 1 \), respectively, and \( nps \) is the net proportionality shift, \( nds \) is the net differential shift and \( nts \) is the net total shift. Other variants are e.g. the shift-share method by Gerfin (Index method), the dynamic shift-share analysis (Barff/Knight 1988) or the extension by Esteban-Marquillas (1972).

As there is more than one way to calculate a Dunn-type shift-share analysis and the terms are not used consequently in the regional economic literature, this function and the documentation use the formulae and terms given in Farhauer/Kroell (2013). If `shift.method = "Dunn"`, this function
shiftp
calculates the net proportionality shift (nps), the net differential shift (nds) and the net total shift (nts) where the last one represents the residuum of (positive) regional factors.

This function calculates an employment prognosis based on a Gerfin shift-share analysis for two years.

Value
A list containing the following objects:

- components: A matrix containing the shift-share components related to the chosen method
- growth: A matrix containing the industry-specific growth values
- prog: A matrix containing the industry-specific prognosis values
- method: The chosen method, e.g. "Dunn"

Author(s)
Thomas Wieland

References


See Also
portfolio, shiftd, shifti, , shift.growth
Examples

# Example data from Spiekermann/Wegener 2008:
# two regions, two industries
region1_2000 <- c(1400, 3600)
region1_2006 <- c(1000, 4400)
region2_2000 <- c(1200, 1800)
region2_2006 <- c(1100, 3700)
region3_2000 <- c(1100, 900)
region3_2006 <- c(800, 1000)
# regional values
nation_2000 <- c(3700, 6300)
nation_2006 <- c(2900, 9100)
# national values
nation_2010 <- c(2500, 12500)
# national prognosis values

# Analysis for region 1:
shiftp(region1_2000, region1_2006, nation_2000,
nation_2006, e_i3 = nation_2010,
time1 = 2000, time2 = 2006, time3 = 2010)
# Analysis for region 2:
shiftp(region2_2000, region2_2006, nation_2000,
nation_2006, e_i3 = nation_2010,
time1 = 2000, time2 = 2006, time3 = 2010)
# Analysis for region 3:
shiftp(region3_2000, region3_2006, nation_2000,
nation_2006, e_i3 = nation_2010,
time1 = 2000, time2 = 2006, time3 = 2010)

Description

This function provides the analysis of regional economic sigma convergence (decline of deviation) for two years using ANOVA (Analysis of Variance)

Usage

sigmaconv(gdp1, time1, gdp2, time2, sigma.measure = "sd",
sigma.log = TRUE, sigma.weighting = NULL, sigma.norm = FALSE,
sigma.issample = FALSE, print.results = FALSE)

Arguments

gdp1 A numeric vector containing the GDP per capita (or another economic variable) at time t
time1 A single value of time t (= the initial year)
gdp2 A numeric vector containing the GDP per capita (or another economic variable) at time $t+1$

time2 A single value of time $t+1$

sigma.measure argument that indicates how the sigma convergence should be measured. The default is $output = \text{"sd"}$, which means that the standard deviation is used. If $output = \text{"var"}$ or $output = \text{"cv"}$, the variance or the coefficient of variation is used, respectively.

sigma.log Logical argument. Per default ($sigma\.log = \text{TRUE}$), also in the sigma convergence analysis, the economic variables are transformed by natural logarithm. If the original values should be used, state $sigma\.log = \text{FALSE}$

sigma.weighting If the measure of statistical dispersion in the sigma convergence analysis (coefficient of variation or standard deviation) should be weighted, a weighting vector has to be stated

sigma.norm Logical argument that indicates if a normalized coefficient of variation should be used instead

sigma.issample logical argument that indicates if the dataset is a sample or the population (default: $is\.sample = \text{FALSE}$, so the denominator of variance is $n$)

print.results Logical argument that indicates if the function shows the results or not

Details

From the regional economic perspective (in particular the neoclassical growth theory), regional disparities are expected to decline. This convergence can have different meanings: Sigma convergence ($\sigma$) means a harmonization of regional economic output or income over time, while beta convergence ($\beta$) means a decline of dispersion because poor regions have a stronger economic growth than rich regions (Capello/Nijkamp 2009). Regardless of the theoretical assumptions of a harmonization in reality, the related analytical framework allows to analyze both types of convergence for cross-sectional data (GDP p.c. or another economic variable, $y$, for $i$ regions and two points in time, $t$ and $t + T$), or one starting point ($t$) and the average growth within the following $n$ years ($t + 1, \ldots, t + n$), respectively. Beta convergence can be calculated either in a linearized OLS regression model or in a nonlinear regression model. When no other variables are integrated in this model, it is called absolute beta convergence. Implementing other region-related variables (conditions) into the model leads to conditional beta convergence. If there is beta convergence ($\beta < 0$), it is possible to calculate the speed of convergence, $\lambda$, and the so-called Half-Life $H$, while the latter is the time taken to reduce the disparities by one half (Allington/McCombie 2007, Goecke/Huether 2016). There is sigma convergence, when the dispersion of the variable ($\sigma$), e.g. calculated as standard deviation or coefficient of variation, reduces from $t$ to $t + T$. This can be measured using ANOVA for two years or trend regression with respect to several years (Furceri 2005, Goecke/Huether 2016).

This function calculates the standard deviation (or variance, coefficient of variation) for the GDP per capita (or another economic variable) for both years and executes an analysis of variance (ANOVA) for these deviation measures (year 1 divided by year 2, F-statistic). If $\sigma_1/\sigma_2 > 0$, there is sigma convergence.
**Value**

Returns a matrix containing the standard deviations, their quotient and the results of the significance test (F-statistic).

**Author(s)**

Thomas Wieland

**References**


**See Also**

rca, sigmaconv.t, betaconv.nls, betaconv.speed, cv, sd2, var2

**Examples**

```r
data(G.counties.gdp)
# Loading GDP data for Germany (counties = Landkreise)

# Using the coefficient of variation

# Using the standard deviation with logged GDP per capita
```
sigmaconv.t  

Analysis of regional sigma convergence for a time series using trend regression

Description

This function provides the analysis of regional economic sigma convergence (decline of deviation) for a time series using a trend regression.

Usage

```r
sigmaconv.t(gdp1, time1, gdp2, time2, sigma.measure = "sd", sigma.log = TRUE,
            sigma.weighting = NULL, sigma.issample = FALSE,
            sigma.plot = FALSE, sigma.plotLSize = 1, sigma.plotLineCol = "black",
            sigma.plotRLine = FALSE, sigma.plotRLineCol = "blue",
            sigma.Ymin = 0, sigma.plotX = "Time", sigma.plotY = "Variation",
            sigma.plotTitle = "Sigma convergence", sigma.bgCol = "gray95",
            sigma.bgrid = TRUE, sigma.bgridCol = "white", sigma.bgridSize = 2,
            sigma.bgridType = "solid", print.results = FALSE)
```

Arguments

- **gdp1**: A numeric vector containing the GDP per capita (or another economic variable) at time \( t \)
- **time1**: A single value of time \( t (= \) the initial year)
- **gdp2**: A data frame containing the GDPs per capita (or another economic variable) at time \( t+1, t+2, t+3, \ldots, t+n \)
- **time2**: A single value of time \( t+1 \)
- **sigma.measure**: argument that indicates how the sigma convergence should be measured. The default is `output = "sd"`, which means that the standard deviation is used. If `output = "var"` or `output = "cv"`, the variance or the coefficient of variation is used, respectively.
- **sigma.log**: Logical argument. Per default (`sigma.log = TRUE`), also in the sigma convergence analysis, the economic variables are transformed by natural logarithm. If the original values should be used, state `sigma.log = FALSE`.
- **sigma.weighting**: If the measure of statistical dispersion in the sigma convergence analysis (coefficient of variation or standard deviation) should be weighted, a weighting vector has to be stated.
- **sigma.issample**: Logical argument that indicates if the dataset is a sample or the population (default: `is.sample = FALSE`, so the denominator of variance is \( n \))
- **sigma.plot**: Logical argument that indicates if a plot of sigma convergence has to be created
- **sigma.plotLSize**: If `sigma.plot = TRUE`: Line size of the sigma convergence plot
sigma.plotLineCol
  If sigma.plot = TRUE: Line color of the sigma convergence plot

sigma.plotRLine
  If sigma.plot = TRUE: Logical argument that indicates if a regression line has to be added to the plot

sigma.plotRLineCol
  If sigma.plot = TRUE and sigma.plotRLine = TRUE: Color of the regression line

sigma.Ymin
  If sigma.plot = TRUE: start value of the Y axis in the plot

sigma.plotX
  If sigma.plot = TRUE: Name of the X axis

sigma.plotY
  If sigma.plot = TRUE: Name of the Y axis

sigma.plotTitle
  If sigma.plot = TRUE: Title of the plot

sigma.bgCol
  If sigma.plot = TRUE: Plot background color

sigma.bgrid
  If sigma.plot = TRUE: Logical argument that indicates if the plot contains a grid

sigma.bgridCol
  If sigma.plot = TRUE and sigma.bgrid = TRUE: Color of the grid

sigma.bgridSize
  If sigma.plot = TRUE and sigma.bgrid = TRUE: Size of the grid

sigma.bgridType
  If sigma.plot = TRUE and sigma.bgrid = TRUE: Type of the grid

print.results
  Logical argument that indicates if the function shows the results or not

Details

From the regional economic perspective (in particular the neoclassical growth theory), regional disparities are expected to decline. This convergence can have different meanings: **Sigma convergence** (σ) means a harmonization of regional economic output or income over time, while **beta convergence** (β) means a decline of dispersion because poor regions have a stronger economic growth than rich regions (Capello/Nijkamp 2009). Regardless of the theoretical assumptions of a harmonization in reality, the related analytical framework allows to analyze both types of convergence for cross-sectional data (GDP p.c. or another economic variable, y, for i regions and two points in time, t and t + T), or one starting point (t) and the average growth within the following n years (t + 1, t + 2, ..., t + n), respectively. Beta convergence can be calculated either in a linearized OLS regression model or in a nonlinear regression model. When no other variables are integrated in this model, it is called **absolute** beta convergence. Implementing other region-related variables (conditions) into the model leads to **conditional** beta convergence. If there is beta convergence (β < 0), it is possible to calculate the speed of convergence, λ, and the so-called Half-Life H, while the latter is the time taken to reduce the disparities by one half (Allington/McCombie 2007, Goecke/Huether 2016). There is **sigma convergence**, when the dispersion of the variable (σ), e.g. calculated as standard deviation or coefficient of variation, reduces from t to t + T. This can be measured using ANOVA for two years or trend regression with respect to several years (Furceri 2005, Goecke/Huether 2016).

This function calculates the standard deviation (or variance, coefficient of variation) for all GDPs per capita (or another economic variable) for the given years and executes a trend regression for these deviation measures. If the slope of the trend regression is negative, there is sigma convergence.
Value

Returns a matrix containing the trend regression model and the resulting significance tests (F-statistic, t-statistic).

Author(s)

Thomas Wieland

References


See Also

rca, sigmaconv, betaconv.nls, betaconv.speed, cv, sd2, var2

Examples

data(G.counties.gdp)
# Loading GDP data for Germany (counties = Landkreise)

# Sigma convergence 2010-2014:
sigmaconv.t (G.counties.gdp$gdppc2010, 2010, G.counties.gdp[65:68], 2014, sigma.plot = TRUE, print.results = TRUE)
# Using the standard deviation with logged GDP per capita

# Using the coefficient of variation (GDP per capita not logged)
Description
Calculating three measures of regional specialization (Gini, Krugman, Hoover) for a set of \( J \) regions

Usage
spec(e_ij, industry.id, region.id, na.rm = TRUE)

Arguments
- \( e_{ij} \) a numeric vector with the employment of the industry \( i \) in region \( j \)
- \( \text{industry.id} \) a vector containing the IDs of the industries \( i \)
- \( \text{region.id} \) a vector containing the IDs of the regions \( j \)
- \( \text{na.rm} \) logical argument that indicates whether NA values should be excluded before computing results

Details
This function is a convenient wrapper for all functions calculating measures of regional specialization (Gini, Krugman, Hoover)

Value
A matrix with three columns (Gini coefficient, Krugman coefficient, Hoover coefficient) and \( J \) rows (one for each regarded region).

Author(s)
Thomas Wieland

References

See Also
gini.spec, krugman.spec2, hoover
Examples

```r
data(G.regions.industries)

spec_j <- spec (e_ij = G.regions.industries$emp_all,
                industry.id = G.regions.industries$ind_code,
                region.id = G.regions.industries$region_code)
```

theil             Theil inequality index

Description

Calculating the Theil inequality index

Usage

```r
theil(x, weighting = NULL, na.rm = TRUE)
```

Arguments

- `x` a numeric vector
- `weighting` a numeric weighting vector, e.g. population
- `na.rm` logical argument that indicates whether NA values should be excluded before computing results

Details

Since there are several Theil measures of inequality, this function uses the formulation from Stoermann (2009).

Value

A single numeric value of the *Theil inequality index* (*0 < TI < 1*).

Author(s)

Thomas Wieland

References


See Also

- `gini`
- `herf`
- `hoover`
Examples

# Example from Stoermann (2009):
regincome <- c(10,10,10,20,50)
theil(regincome)
# 0.2326302

to.dummy

Creating dummy variables

Description

This function creates a dataset of dummy variables based on an input character vector.

Usage

to.dummy(x)

Arguments

x
A character vector

Details

This function transforms a character vector x with c characteristics to a set of c dummy variables whose column names corresponding to these characteristics marked with "_DUMMY".

Value

A data.frame with dummy variables corresponding to the levels of the input variable.

Note

This function contains code from the authors’ package MCI.

Author(s)

Thomas Wieland

References


Examples

charvec <- c("Peter", "Paul", "Peter", "Mary", "Peter", "Paul")
# Creates a vector with three names (Peter, Paul, Mary)
to.dummy(charvec)
# Returns a data frame with 3 dummy variables
# (Mary_DUMMY, Paul_DUMMY, Peter_DUMMY)
Description
Calculating the variance (var), weighted or non-weighted, for samples or populations

Usage
\texttt{var2(x, is.sample = TRUE, weighting = NULL, wmean = FALSE, na.rm = TRUE)}

Arguments
\begin{itemize}
  \item \texttt{x} \hspace{1cm} a numeric vector
  \item \texttt{is.sample} \hspace{1cm} logical argument that indicates if the dataset is a sample or the population (default: \texttt{is.sample = TRUE}, so the denominator of variance is \(n - 1\))
  \item \texttt{weighting} \hspace{1cm} a numeric vector containing weighting data to compute the weighted standard deviation (instead of the non-weighted sd)
  \item \texttt{wmean} \hspace{1cm} logical argument that indicates if the weighted mean is used when calculating the weighted standard deviation
  \item \texttt{na.rm} \hspace{1cm} logical argument that whether NA values should be extracted or not
\end{itemize}

Details
The function calculates the variance (var). Unlike the R base \texttt{var} function, the \texttt{var2} function allows to choose if the data is treated as sample (denominator of variance is \(n - 1\)) or not (denominator of variance is \(n\)).

From a regional economic perspective, var and sd is closely linked to the concept of \textit{sigma convergence} (\(\sigma\)) which means a harmonization of regional economic output or income over time, while the other type of convergence, \textit{beta convergence} (\(\beta\)), means a decline of dispersion because poor regions have a stronger growth than rich regions (Capello/Nijkamp 2009). The sd allows to summarize regional disparities (e.g. disparities in regional GDP per capita) in one indicator. The coefficient of variation (see the function \texttt{cv}) is more frequently used for this purpose (e.g. Lessmann 2005, Huang/Leung 2009, Siljak 2015). But the sd can also be used for any other types of disparities or dispersion, such as disparities in supply (e.g. density of physicians or grocery stores).

The variance can be weighted by using a second weighting vector. As there is more than one way to weight measures of statistical dispersion, this function uses the formula for the weighted variance (\(\sigma_w\)) from Sheret (1984). The vector \(x\) is automatically treated as a sample (such as in the base \texttt{sd} function), so the denominator of variance is \(n - 1\), if it is not, set \texttt{is.sample = FALSE}.

Value
Single numeric value. If weighting is specified, the function returns a weighted variance (optionally using a weighted arithmetic mean if \texttt{wmean = TRUE}).
Author(s)
Thomas Wieland

References
Bahrenberg, G./Giese, E./Mevenkamp, N./Nipper, J. (2010): “Statistische Methoden in der Geogra-
twenty-first century - recent theoretical advances and future challenges”. In: Capello, R./Nijkamp,
pdf.
Huang, Y./Leung, Y. (2009): “Measuring Regional Inequality: A Comparison of Coefficient of
Variation and Hoover Concentration Index”. In: The Open Geography Journal, 2, p. 25-34.
Research, 15, 3, p. 289-295.
Siljak, D. (2015): “Real Economic Convergence in Western Europe from 1995 to 2013”. In: Interna-

See Also
sd2, cv, gini, herf, hoover, mean2, rca

Examples
# Regional disparities / sigma convergence in Germany
data(G.counties.gdp)
# GDP per capita for German counties (Landkreise)
vars <- apply(G.counties.gdp[54:68], MARGIN = 2, FUN = var2)
# Calculating variance for the years 2000-2014
years <- 2000:2014
plot(years, vars, "l", xlab = "year",
ylab = "Variance of GDP per capita")
# Plot variance over time

williamson  Williamson index

Description
Calculating the Williamson index (population-weighted coefficient of variation)

Usage
williamson (x, weighting, coefnorm = FALSE, wmean = FALSE, na.rm = TRUE)
Arguments

- **x**: a numeric vector
- **weighting**: mandatory: a numeric vector containing weighting data (usually regional population)
- **coefnorm**: logical argument that indicates if the function output is the standardized cv ($0 < v* < 1$) or not ($0 < v < \infty$) (default: coefnorm = FALSE)
- **wmean**: logical argument that indicates if the weighted mean is used when calculating the weighted coefficient of variation
- **na.rm**: logical argument that whether NA values should be extracted or not

Details

The *Williamson index* (Williamson 1965) is a population-weighted coefficient of variation.

The *coefficient of variation*, $v$, is a dimensionless measure of statistical dispersion ($0 < v < \infty$), based on variance and standard deviation, respectively. The $cv$ (variance, standard deviation) can be weighted by using a second weighting vector. As there is more than one way to weight measures of statistical dispersion, this function uses the formula for the weighted $cv$ ($v_w$) from Sheret (1984). The $cv$ can be standardized, while this function uses the formula for the standardized $cv$ ($v*$, with $0 < v* < 1$) from Kohn/Oeztuerk (2013). The vector $x$ is automatically treated as a sample (such as in the base `sd` function), so the denominator of variance is $n - 1$, if it is not, set `is.sample = FALSE`.

Value

Single numeric value. If coefnorm = FALSE the function returns the non-standardized $cv$ ($0 < v < \infty$). If coefnorm = TRUE the standardized $cv$ ($0 < v* < 1$) is returned.

Author(s)

Thomas Wieland

References


Huang, Y./Leung, Y. (2009): “Measuring Regional Inequality: A Comparison of Coefficient of Variation and Hoover Concentration Index”. In: *The Open Geography Journal, 2*, p. 25-34.


See Also
gini, herf, hoover, cv, disp

Examples
data(GoettingenHealth2)
# districts with healthcare providers and population size

williamson((GoettingenHealth2$phys_gen/GoettingenHealth2$pop), GoettingenHealth2$pop)
Index

atkinson, 4, 17, 23, 68
Automotive, 5

betaconv.nls, 6, 11, 13, 100, 123, 126
betaconv.ols, 9, 9, 13, 100
betaconv.speed, 9, 11, 12, 100, 123, 126

conc, 14
converse, 15, 64, 73, 102
coulter, 5, 16, 22, 23, 68
curvefit, 17
cv, 5, 9, 11, 13, 17, 19, 22, 23, 49, 56, 66, 68, 91, 93, 100, 123, 126, 131, 133
dalton, 5, 17, 21, 22, 23
disp, 5, 17, 22, 22, 68, 133
dist, 24, 27, 104
dist.buf, 23, 25, 27, 64, 104
dist.calc, 25, 64
dist.mat, 24, 25, 26, 64, 104
durpug, 27
dalton, 5, 17, 21, 22, 23
disp, 5, 17, 22, 22, 68, 133
dist, 24, 27, 104
dist.buf, 23, 25, 27, 64, 104
dist.calc, 25, 64
dist.mat, 24, 25, 26, 64, 104
durpug, 27
eellison.a, 28, 30–32, 80, 81, 88
ellison.a2, 29, 31, 32, 80, 81, 88
ellison.c, 30, 32, 69–71, 79, 80, 81, 88
ellison.c2, 31, 31, 69–71, 80, 81, 88
EU28.emp, 33
FK2014_EGC, 34
Freiburg, 35
G.counties.gdp, 36
G.regions.emp, 39
G.regions.industries, 40
gifpro, 41, 46
gifpro.tbs, 43, 44
gini, 5, 17, 21, 22, 46, 52, 54, 56, 68, 106, 128, 131, 133
gini.conc, 15, 29–32, 49, 50, 54, 56, 75–77, 79–81, 84, 88, 91
gini.spec, 28–32, 49, 52, 56, 75–77, 79–81, 84, 88, 91, 127
gini2, 5, 17, 22, 23, 55
Goettingen, 57
GoettingenHealth1, 59
GoettingenHealth2, 60
growth, 61
hansen, 62, 73
herf, 5, 17, 21–23, 49, 56, 65, 68, 91, 106, 128, 131, 133
howard.c1, 29–32, 68, 70, 71
howard.xcl, 29–32, 69, 69
howard.xcl2, 29–32, 69, 70, 70, 71
huff, 16, 64, 71, 102
krugman.conc, 74, 76, 77, 79
krugman.conc2, 15, 75, 75, 77, 79
krugman.spec, 28, 75, 76, 77, 79
krugman.spec2, 75–77, 78, 127
litzenberger, 29–32, 79, 81, 88
litzenberger2, 29–32, 80, 80
lm.beta, 82
locq, 29–32, 75–77, 79–81, 83, 87, 88
locq.growth, 85
locq2, 29–32, 80, 81, 84, 87
lorenz, 89
mean2, 92, 106, 131
mssd, 93
portfolio, 43, 46, 87, 94, 109, 111, 113, 115, 126, 131
rca, 9, 11, 21, 96, 106, 123, 126, 131
REAT (REAT-package), 3
REAT-package, 3
reilly, 16, 64, 73, 101
INDEX

ripley, 103

sd2, 9, 11, 13, 23, 92, 93, 100, 105, 123, 126, 131

shift, 43, 46, 87, 96, 107, 111, 113, 115, 118
shift.growth, 109, 109, 113, 115, 118, 120
shiftd, 43, 46, 87, 96, 109, 111, 111, 120
shifti, 43, 46, 87, 96, 109, 111, 113, 114, 115, 118, 120
shiftid, 116
shiftp, 118
sigmaconv, 9, 11, 13, 100, 121, 126
sigmaconv.t, 9, 11, 13, 100, 123, 124
spec, 127

theil, 5, 17, 22, 23, 68, 128
to.dummy, 129

var2, 9, 11, 13, 93, 100, 123, 126, 130

williamson, 23, 131