# Package ‘REAT’

May 4, 2017

**Type**  Package  
**Title**  Regional Economic Analysis Toolbox  
**Version**  1.3.2  
**Date**  2017-05-04  
**Author**  Thomas Wieland  
**Maintainer**  Thomas Wieland <thomas.wieland.geo@googlemail.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**  2017-05-04 07:14:02 UTC  

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAT-package</td>
<td>2</td>
</tr>
<tr>
<td>converse</td>
<td>3</td>
</tr>
<tr>
<td>cv</td>
<td>5</td>
</tr>
<tr>
<td>data.dummy</td>
<td>7</td>
</tr>
<tr>
<td>data.index</td>
<td>8</td>
</tr>
<tr>
<td>disp</td>
<td>9</td>
</tr>
<tr>
<td>dist.buf</td>
<td>10</td>
</tr>
<tr>
<td>dist.calc</td>
<td>12</td>
</tr>
<tr>
<td>dist.mat</td>
<td>13</td>
</tr>
<tr>
<td>Freiburg</td>
<td>14</td>
</tr>
<tr>
<td>G.counties.gdp</td>
<td>15</td>
</tr>
<tr>
<td>G.regions.emp</td>
<td>18</td>
</tr>
<tr>
<td>gini</td>
<td>20</td>
</tr>
<tr>
<td>gini.conc</td>
<td>23</td>
</tr>
<tr>
<td>gini.spec</td>
<td>26</td>
</tr>
</tbody>
</table>
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 in seven groups:

(1) analysis of regional disparities and inequality, including Gini coefficient, the Lorenz curve and the (weighted) coefficient of variation

(2) specialization of regions, including spatial Gini coefficient of regional specialization and Krugman coefficient for regional specialization

(3) spatial concentration of industries, including location quotients and spatial Gini coefficient for industry concentration

(4) regional growth and convergence, including traditional shift-share analysis and analysis of beta and sigma convergence for cross-sectional data

(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 and visualization.
The package also contains data examples.

Author(s)

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

References


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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_a</td>
<td>a single numeric value of attractivity/population size of location/city a</td>
</tr>
<tr>
<td>P_b</td>
<td>a single numeric value of attractivity/population size of location/city b</td>
</tr>
<tr>
<td>D_ab</td>
<td>a single numeric value of the transport costs (e.g. distance) between a and b</td>
</tr>
</tbody>
</table>
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)
Coefficient of variation

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

Usage

\[
\text{cv}(x, \text{is.sample} = \text{TRUE}, \text{coefnorm} = \text{FALSE}, \text{weighting} = \text{NULL}, \text{wmean} = \text{FALSE}, \text{na.rm} = \text{FALSE})
\]

Arguments

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

```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
```
**Description**

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

**Usage**

```r
data.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**

```r
charvec <- c("Peter", "Paul", "Peter", "Mary", "Peter", "Paul")
# Creates a vector with three names (Peter, Paul, Mary)
data.dummy(charvec)
# Returns a data frame with 3 dummy variables
# (Mary_DUMMY, Paul_DUMMY, Peter_DUMMY)
```
Description

Standardizing a variable as an index

Usage

data.index(dataset, col.index, col.ref, value.ref)

Arguments

dataset regarded as data.frame
col.index column to be converted to index values
col.ref column with the reference values of the index (e.g. years or months)
value.ref value from the reference column which is the reference value (=100)

Value

A numeric vector consisting of the indexed values of col_index

Author(s)

Thomas Wieland

Examples

# Creating test data
cyear <- 2010:2015
cvalues <- c(20,24,21,28,27,29)
timeseries <- data.frame(year, values)
timeseries <- data.frame(year, values)
data.index(timeseries, "values", "year", "2012")
data.index(timeseries, "values", "year", "2012")
# returns index values
data.index(timeseries, "values", "year", "2012")
timeseries <- data.frame(timeseries, values)
timeseries <- data.frame(timeseries, values)
# add index values to data
Description

Calculating the Gini coefficient (non-standardized and standardized), the Herfindahl-Hirschman coefficient (non-standardized and standardized) and the Herfindahl-Hirschman equivalent number and the coefficient of variation (non-standardized and standardized)

Usage

disp(x)

Arguments

x a numeric vector containing the regarded objects

Details

The Gini coefficient and the Herfindahl-Hirschman coefficient are measures of the degree of a concentration (e.g. household income, sales or market shares of firms in an industry, distribution of facilities in regions). The coefficient of variation is a simple standardized measure of distribution. This function returns these coefficients as non-standardized ($G$, $HHI$, $CV$) and standardized values ($G^*$, $HHI^*$, $CV^*$) and the HHI equivalent number ($HHI_{eq}$). For more information about the coefficients, see the single function documentations (gini, herf, herf_eq and cv).

Value

a list with the 7 entries (=result values):

- HHI Herfindahl-Hirschman coefficient, non-standardized
- HHI_n Herfindahl-Hirschman coefficient, standardized
- HHI_eq Herfindahl-Hirschman equivalent number
- GINI Gini coefficient, non-standardized
- GINI_n Gini coefficient, standardized
- CV Coefficient of variation, non-standardized
- CV_n Coefficient of variation, standardized

Author(s)

Thomas Wieland
References

Bahrenberg, G./Giese, E./Mevenkamp, N./Nipper, J. (2010): “Statistische Methoden in der Geogra-


See Also

`gini, herf, cv`

Examples

```r
# Example from Doersam (2004)
# (Sales of four car manufacturing firms)
sales <- c(20,50,20,10)
disp(sales)
```

dist.buf

*Counting points in a buffer*

Description

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

Usage

```r
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

```r
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 data.frame containing 2 columns: The start point IDs (from) and the number of counted points in the given buffer distance (count_location).

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

Examples

citynames <- c("Goettingen", "Karlsruhe", "Freiburg")
lat <- c(51.556307, 49.009603, 47.9874)
lon <- c(9.947375, 8.417004, 7.8945)
citynames <- c("Goettingen", "Karlsruhe", "Freiburg")
cities <- data.frame(citynames, lat, lon)
dist.mat (cities, "citynames", "lat", "lon", cities, "citynames", "lat", "lon")
# Euclidean distance matrix (3 x 3 cities = 9 distances)
dist.buf (cities, "citynames", "lat", "lon", cities, "citynames", "lat", "lon", bufdist = 300000)
# Cities within 300 km
dist.calc  

Euclidean distance between coordinates

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

Usage

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

dist.calc(51.556307, 9.947375, 49.009603, 8.417004)  
# about 304 kilometers
Dist.mat

Euclidean distance matrix between points

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

Usage
```
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
See Also

dist, dist.buf

Examples

citynames <- c("Goettingen", "Karlsruhe", "Freiburg")
lat <- c(51.556307, 49.009663, 47.9874)
lon <- c(9.947375, 8.417004, 7.8945)
citynames <- c("Goettingen", "Karlsruhe", "Freiburg")
cities <- data.frame(citynames, lat, lon)
dist.mat(cities, "citynames", "lat", "lon")
cities, "citynames", "lat", "lon")
# Euclidean distance matrix (3 x 3 cities = 9 distances)
dist.buf(cities, "citynames", "lat", "lon", cities, "citynames", "lat", "lon", bufdist = 300000)
# Cities within 300 km

Freiburg
Employment data in Freiburg and Germany

Description

Dataset with industry-specific employment in Freiburg and Germany in the years 2008 and 2014

Usage

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

data(Freiburg)
# Loads the data
industries <- Freiburg$industry
x <- Freiburg$e_g_Freiburg_0814
y <- Freiburg$e_g_Germany_0814
z <- Freiburg$e_Freiburg2014
portfolio(x,y,z, "Freiburg", "Germany", "Growth portfolio Freiburg and Germany",
pcol="given", colsp=Freiburg$color, leg=1, leg_vec=industries, leg_fsize=0.6)
# Creates a portfolio comparing the industry growth in Freiburg and Germany

G.counties.gdp  Gross Domestic Product (GDP) per capita for German counties 1992-2014

Description

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
G.counties.gdp

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
G.counties.gdp

- 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

Details

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


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
```

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 == "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

# 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

# Concentration of financial industry in Germany 2014
# based on 16 German regions (Bundeslaender) for 2014
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",]
gini <- function(x, coefnorm = FALSE, weighting = NULL, lc = FALSE, 
   lcx = "% of objects", lcy = "% of regarded variable", 
   lctitle = "Lorenz curve", le.col = "blue", lc.col = "black", 
   lsize = 1, ltype = "solid", 
   bg.col = "gray95", bgrid = TRUE, bgrid.col = "white", 
   bgrid.size = 2, bgrid.type = "solid", 
   lcg = FALSE, lcgN = FALSE, lcgNcaption = NULL, 
   lcgNlabNx = 0, lcgNlabNy = 1, add.lc = FALSE)

Arguments

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

coeform
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)

lc
logical argument that indicates if the Lorenz curve is plotted additionally (de-
fault: 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)
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", 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

```r
gini.conc <- c(1000, 1200, 1600, 400, 200, 600)
```

# size of income classes

```r
gini (income, weighting = sizeofclass)
```

# returns the non-standardized Gini coefficient (0.5278)

# Regional disparities in Germany:

```r
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)

```r
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**

```r
gini.conc(e_ij, e_j, lc = FALSE, lcx = "% of objects",
          lcy = "% of regarded variable", lctitle = "Lorenz curve",
          le.col = "blue", lc.col = "black", lsize = 1, ltype = "solid",
          bg.col = "gray95", bgrid = TRUE, bgrid.col = "white",
          bgrid.size = 2, bgrid.type = "solid", lcg = FALSE, lcgn = FALSE,
          lcg.caption = NULL, lcg.lab.x = 0, lcg.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
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 `lc = 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 == "Insgesamt") & G.regions.emp$year == "2008",]
# only data for construction industry (Baugewerbe) and all-over (Insgesamt)
# for the 16 German regions in the year 2008
construction2008 <- construction2008[construction2008$region != "Insgesamt",]
# delete all-over data for all industries
gini.conc(construction2008[construction2008$industry=="Baugewerbe (F)",]$emp, construction2008[construction2008$industry=="Insgesamt",]$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 == "Insgesamt") & G.regions.emp$year == "2008",]
finance2014 <- G.regions.emp[(G.regions.emp$industry == "Erbringung von Finanz- und Vers.leistungen (K)" | G.regions.emp$industry == "Insgesamt") & G.regions.emp$year == "2014",]
Description

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

Usage

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

Arguments

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

**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

**lcg.caption** if lcg = TRUE (displaying the Gini coefficient in the plot), lcg.caption specifies the caption above the coefficients

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

**lcg.lab.y** if lcg = TRUE (displaying the Gini coefficient in the plot), lcg.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 \( \text{gini}() \)). 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)

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)

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)
dist the column in the interaction matrix od_dataset containing the transport costs (e.g. travelling time, distance)
gamma a single numeric value for the exponential weighting ($\gamma$) of size (default: 1)
lambda a single numeric value for the exponential weighting ($\lambda$) of distance (transport costs, default: -2)
atypet Type of attractivity weighting function: atype = "pow" (power function),
atype = "exp" (exponential function) or atype = "logistic" (default: atype = "pow")
dtypte 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
dist_const a numeric value of a constant to be added to the transport costs (e.g. 1)
dist_max a numeric value of a maximal value of transport costs for the opportunities to be recognized
extract_local logical argument that indicates if the start points should be included in the analysis or not (if \( i = j \)). Default value: extract_local = FALSE
acccnorm logical argument that indicates if the Hansen accessibility should be standardized
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

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

Returns a data frame with the origins and the accessibility values (column accessibility).

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
save(shoppingcenters1, file="shoppingcenters1.rda")
huff.shares <- huff(shoppingcenters1, "communities", "locations", "S_j", "d_ij")
# Market shares of the new location:
huff.shares[huff.shares$locations == "New Store",]
# Hansen accessibility for Oak Hammock and Rock Creek:
hansen (huff.shares, "communities", "locations", "S_j", "d_ij")
Description
Calculating the Herfindahl-Hirschman coefficient of concentration, standardized and non-standardized

Usage
herf(x, coefnorm = FALSE, output = "HHI")

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

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* ($H_{eq} >= 1$).

Author(s)
Thomas Wieland
References


See Also

cv, gini

Examples

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

# Regional disparities in Germany:
gdp <- c(468.69, 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)
erf(gdp) # returns the HHI (0.125)
```

---

**hoover**

*Hoover Concentration Index*

**Description**

Calculating the Hoover Concentration Index with respect to regional income (e.g. GDP) and population

**Usage**

`hoover(x, weighting = NULL)`
Arguments

- \( \text{x} \)  
  A numeric vector (dataset of regional income, e.g. GDP)

- \( \text{weighting} \)  
  A numeric weighting vector (dataset of regional population). If \( \text{weighting} = \text{NULL} \), the shares of income are compared with the shares of regions \( (1/n) \)

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


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

See Also

cv, gini, herf

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
pop <- 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)
**huff**

**Huff model**

**Description**

Calculating market areas using the probabilistic market area model by Huff

**Usage**

```r
huff(huffdataset, origins, locations, attrac, dist, gamma = 1, lambda = -2,
atype = "pow", dtype = "pow", gamma2 = NULL, lambda2 = NULL, output = "shares",
localmarket_dataset = NULL, origin_id = NULL, localmarket = NULL, check_df = TRUE)
```

**Arguments**

- **huffdataset**
  an interaction matrix which is a DataFrame 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 attractivity 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 attractivity 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 γ parameter is needed

- **lambda2**
  if dtype = "logistic" a second λ parameter is needed

- **output**
  argument that indicates the type of function output: if output = "shares", the Huff function returns an interaction/probability matrix), if output = "total", the function returns the total sales of the locations. Default: output = "shares"

- **localmarket_dataset**
  if output = "total", a DataFrame 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: \(U_{ij} = A_j^\gamma d_{ij}^{-\lambda}\). 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. If \texttt{output = "shares"}, the function returns an estimated interaction matrix. If \texttt{output = "total"} you need local market information about the origins (e.g. purchasing power, population size etc.) filed in another data.frame and the function results are the total sales/shares of the given stores/locations. Note that each attractivity or distance value must be greater than zero.

Value

Returns either the input interaction matrix including the calculated shares \((p_{-i,j})\) (if \texttt{output = "shares"}) or the total sales \((\text{sum}_E_{-j})\) and total shares \((\text{share}_j)\) of the stores locations (if \texttt{output = "total"}). Both results are data.frame.

Note

This function contains code from the authors’ package MCI.

Author(s)

Thomas Wieland

References


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
save(shoppingcenters1, file="shoppingcenters1.rda")
huff_shares <- huff(shoppingcenters1, "communities", "locations", "S_j", "d_ij")
# Market shares of the new location:
huff_shares[huff_shares$locations == "New Store",]
# Hansen accessibility for Oak Hammock and Rock Creek:
hansen (huff_shares, "communities", "locations", "S_j", "d_ij")

# Example from Berman/Evans (2012):
locations <- c(1, 2, 3)
S_j <- c(200, 300, 500)
location_data <- data.frame(locations, S_j)
d_ij <- c(7, 10, 15)
interactionmatrix <- data.frame(location_data, d_ij)
interactionmatrix$cgroupl <- 1
shoppingcenters2 <- interactionmatrix
huff (shoppingcenters2, "cgroupl", "locations", "S_j", "d_ij")

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)
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
krugman.spec

Examples
# 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.spec

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
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

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)

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).
lm.beta

Value

A single numeric value (0 < K_{ji} < 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

---

*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 $\beta$ 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 $\beta$ 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

```r
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*
Usage

locq(e_ij, e_j, e_i, e)

Arguments

e_ij  a single numeric value with the employment of industry i in region j

Arguments

e_ij  a single numeric value with the employment of industry i in region j

e_j  a single numeric value with the over-all employment in region j

Arguments

e_i  a single numeric value with the over-all employment in industry i

e  a single numeric value with the over-all employment in all regions

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. 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 (\( LQ \))

Author(s)

Thomas Wieland

References


See Also

gini.conc, gini.spec

Examples

# Example from Farhauer/Kroell (2013):
locq (1714, 79086, 879213, 15593224)
# returns the location quotient (0.3847623)
**Description**

Calculating and plotting the Lorenz curve

**Usage**

```r
lorenz(x, weighting = NULL, z = NULL,
    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, lcgn = 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
- **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
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

```r
# 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",
ltitle = "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, 35.03, 112.66, 56.22, 85.61, 56.81)
# GDP of german regions 2015 (in billion EUR)
lorenz(gdp, lcg = TRUE, lcgn = TRUE)
# plots the Lorenz curve with user-defined title and labels
# including Gini coefficient
```

### mean2

**Calculation of mean (extended)**

**Description**

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

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

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

Bahrenberg, G./Giese, E./Mevenkamp, N./Nipper, J. (2010): “Statistische Methoden in der Geogra-

Research, 15, 3, p. 289-295.

See Also

sd2

Examples

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")
Portfolio matrix

Description

Portfolio matrix plot comparing two numeric vectors

Usage

```r
portfolio(x, y, z, label_x = "X", label_y = "Y", heading = "Portfolio",
  pcol = "given", colsp = 0, leg = FALSE, leg_vec = 0, leg_fsize = 1,
  leg_x = -max_val, leg_y = -max_val/2)
```

Arguments

- **x**: A numeric vector representing the values for the x axis
- **y**: A numeric vector representing the values for the y axis
- **z**: A numeric vector representing the size of the points/bubbles
- **label_x**: Label for the x axis
- **label_y**: Label for the y axis
- **heading**: Heading for the plot
- **pcol**: Indicates if the colors of the points are given by the user (pcol = "given") and defined by the vector colsp or set by random (pcol = "random")
- **colsp**: A vector representing the user-defined colors of the points
- **leg**: Logical argument that indicates if the plot has a legend or not (default: leg = FALSE)
- **leg_vec**: If leg = TRUE, this vector defines the values for the plot legend
- **leg_fsize**: If leg = TRUE, this value defines the font size of the legend
- **leg_x**: If leg = TRUE: x coordinate for the legend (default: leg_x=-max_val, where max_val is the maximum value of all values in the dataset)
- **leg_y**: If leg = TRUE: y coordinate for the legend (default: leg_y=-max_val/2, where max_val is the maximum value of all values in the dataset)

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 plot of the portfolio matrix

Author(s)

Thomas Wieland

References


See Also

shift

Examples

data(Freiburg)
# Loads the data
industries <- Freiburg$industry
x <- Freiburg$e_g_Freiburg_0814
y <- Freiburg$e_g_Germany_0814
z <- Freiburg$e_Freiburg2014
portfolio(x,y,z, "Freiburg", "Germany", "Growth portfolio Freiburg and Germany",
          pcol="given", colsp=Freiburg$color, leg=1, leg_vec=industries, leg_fsize=8.6)
# Creates a portfolio comparing the industry growth in Freiburg and Germany

rca  Analysis of regional convergence

Description

This function provides the analysis of absolute regional economic convergence (beta and sigma convergence) for cross-sectional data.

Usage

rca(gdp1, time1, gdp2, time2, output = "all", sigma.measure = "cv",
     sigma.log = TRUE, sigma.norm = FALSE, sigma.weighting = NULL, digs = 5)
**Arguments**

- **gdp1**: A numeric vector containing the GDP per capita (or another economic variable) at time $t$.
- **time1**: A single value of time $t$, e.g., 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 $t+1$.
- **output**: Argument that indicates the type of function output: if output = "all" (default), the function returns a list containing the results. If output = "data", the function only returns the input variables and their transformations in a data.frame. If output = "lm", an lm object of the (linearized) model is returned.
- **sigma.measure**: Argument that indicates how the sigma convergence should be measured. The default is output = "cv", which means that a coefficient of variation is used. If output = "sd", the standard deviation is used.
- **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.norm**: Logical argument that indicates if a normalized coefficient of variation should be used instead.
- **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.
- **digs**: The number of digits for the resulting values (default: digs = 5).

**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 $t+T$). Given two GDPs per capita or another economic variable, ($y_i$ for $i$ regions) and the related two points in time ($t$ and $t+T$), if there is beta convergence ($-1 < \beta < 0$), 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). 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 $t+T$ (Furceri 2005).

This function needs two vectors (GDP p.c. or another economic variable, $y_i$ for $i$ regions) and the related two points in time ($t$ and $t+T$). If output = "all", it returns the estimation results of beta convergence and, if $-1 < \beta < 0$, also the calculations of $\lambda$ and $H$ related to $\beta$. The *sigma convergence* is operationalized as the difference between the dispersions of the regarded variable (in-transformed if sigma.log = TRUE): $\sigma_t - \sigma_{t+T}$. If this value is positive, there is *sigma convergence.*
with respect to these points in time. The dispersions can be calculated as (weighted or non-weighted, standardized or non-standardized) standard deviation or coefficient of variation (see the function cv), to be stated by the function parameters sigma.measure, sigma.norm and sigma.weighting. State output = "lm" for the underlying regression model (lm object) only or output = "data" for the transformed dataset. As yet, the function only allows absolute beta convergence.

Value

If output = "all": a list containing the items

constant The constant in the beta convergence OLS model
beta The "slope" of the OLS model (beta convergence)
tinterval Time interval between t and t+T, in units of time
lambda Lambda, the speed of convergence (NA in absence of beta convergence)
halflife H, the half-life value (NA in absence of beta convergence)
r.squared R-Squared of the OLS model
N Number of regarded regions
sigma Difference in dispersion parameter between t and t+T

If output = "data": a data.frame containing the columns

gdp1 the input GDP per capita (or another economic variable) at time t
gdp2 the input GDP per capita (or another economic variable) at time t+T
diff the absolute difference between gdp2 and gdp1 ((t+T) - t)
diff the relative difference between gdp2 and gdp1 ((t+T) - t)
ln_growth natural logarithm of the growth
ln_initial natural logarithm of the initial value at time t

If output = "lm": A lm object of the estimated OLS model

Author(s)

Thomas Wieland

References


See Also

cv

Examples

```r
# Regional disparities / beta and sigma convergence in Germany
data(G.counties.gdp)
# GDP per capita for German counties (Landkreise)
# returns a list
beta <- convergence$beta
# Beta convergence value
```

---

Description

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

Usage

```r
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

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 = FALSE)

Arguments

x
is.sample
weighting
wmean
na.rm

a numeric vector
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)
a numeric vector containing weighting data to compute the weighted standard deviation (instead of the non-weighted sd)
logical argument that indicates if the weighted mean is used when calculating the weighted standard deviation
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 (σ) which means a harmonization of regional economic output or income over time, while the other type of convergence, beta convergence (β), 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
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_gdp < - apply (G.counties.gdp[54:68], MARGIN = 2, FUN = sd)
# Calculating standard deviation for the years 2000-2014
years < - 2000:2014
# vector of years (2000-2014)
plot(years, sd_gdp, "l", ylim = c(0,15000), xlab = "Year",
ylab = "SD of GDP per capita")
# Plot sd over time
shift  

Shift-share analysis

Description
Analyzing regional growth with the shift-share analysis

Usage
`shift(region_t, region_t1, nation_t, nation_t1)`

Arguments
- `region_t`: a numeric vector with `i` values containing the employment in `i` industries in a region at time `t`
- `region_t1`: a numeric vector with `i` values containing the employment in `i` industries in a region at time `t + 1`
- `nation_t`: a numeric vector with `i` values containing the employment in `i` industries in the national economy at time `t`
- `nation_t1`: a numeric vector with `i` values containing the employment in `i` industries in the national economy at time `t + 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.

As there is more than one way to calculate a shift-share analysis and the terms are not used consistently in the regional economic literature, this function and the documentation use the formulae and terms given in Farhauer/Kroell (2013). 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.

Value
- `nps`: The net proportionality shift
- `nds`: The net differential shift
- `nts`: The net total shift
Author(s)

Thomas Wieland

References


See Also

portfolio

Examples

```r
# 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(50,30,30,40)
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)
```
Description

Calculating the Theil inequality index

Usage

theil(x)

Arguments

x         a numeric vector

Details

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

Value

A single numeric value of the Hoover Concentration Index (0 < CI < 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
Index

converse, 3, 30, 36, 53

cv, 5, 10, 22, 32, 33, 46, 52

sd2, 47, 54

shift, 49, 56

data.dummy, 7

data.index, 8

disp, 9

dist, 11, 14

dist.buf, 10, 12, 14, 30

dist.calc, 12, 30

dist.mat, 11, 12, 13, 30

Freiburg, 14

G.counties.gdp, 15

G.regions.emp, 18

gini, 6, 10, 20, 25, 27, 32, 33, 55, 58

gini.conc, 22, 23, 27, 37, 38, 40, 41, 43, 46

gini.spec, 22, 25, 26, 37, 38, 40, 41, 43, 46

hansen, 28

herf, 6, 10, 22, 31, 33, 46, 55, 58

hoover, 6, 22, 32, 46, 55, 58

huff, 4, 30, 34, 53

krugman.conc, 36, 38, 40, 41

krugman.conc2, 37, 38, 40, 41

krugman.spec, 37, 38, 39, 41

krugman.spec2, 37, 38, 40, 40

lm.beta, 41

locq, 37, 38, 40, 41, 42

lorenz, 44

mean2, 46, 55

portfolio, 48, 57

rca, 6, 49, 55

REAT (REAT-package), 2

REAT-package, 2

reilly, 4, 30, 36, 52