Package ‘REAT’

October 13, 2018

Type Package
Title Regional Economic Analysis Toolbox
Version 2.0.3
Date 2018-10-13
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
Depends R (>= 2.10)
Repository CRAN
Date/Publication 2018-10-13 11:40:03 UTC

R topics documented:

REAT-package .................................................. 2
betaconv.nls .................................................. 4
betaconv.ols .................................................. 6
betaconv.speed .............................................. 10
converse .................................................... 11
cv ............................................................. 13
dist.buf ....................................................... 15
dist.calc ..................................................... 16
dist.mat ....................................................... 17
EU28.emp ..................................................... 18
Freiburg ....................................................... 19
G.counties.gdp ............................................... 20
G.regions.emp ............................................... 23
gini .......................................................... 24
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, Guesefeldt 1999). The group of the related analysis methods is sometimes summarized by the term regional analysis or regional economic analysis (Dinc 2015, Guesefeldt 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 und visualization.

The package also contains data examples.

Author(s)

Thomas Wieland

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

References


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", output.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 \( t (= \text{the initial year}) \)
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 \( t+1, t+2, t+3, ..., t+n \)
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
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)
# 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", output.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 \( t+1, t+2, t+3, \ldots, t+n \)
time2 A single value of time \( t+1 \) or \( t_n \), respectively
conditions A data frame containing the conditions for conditional beta convergence
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
output.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

```r
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
betaconv1$cbeta$estimates
# Addressing estimates for the conditional beta model

# Three years (2010-2012), no conditions (Absolute beta convergence)

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

betaconv2 <- betaconv.ols (G.counties.gdp$gdppc2010, 2010, G.counties.gdp[65:66], 2012, conditions = G.counties.gdp[c(70,71)], output.results = TRUE)
# Store results in object
betaconv2$cbeta$estimates
# Addressing estimates for the conditional beta model
```
### 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, output.results = TRUE)
```

**Arguments**

- `beta` (Beta value)
- `tinterval` (Time interval (in time units, such as years))
- `output.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 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

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

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)

Coefficient of variation

Description

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

Usage

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

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 \(\text{FALSE}\).

Value

Single numeric value. If coefnorm \(\text{FALSE}\) the function returns the non-standardized cv \((0 < v < \infty\). If coefnorm \(\text{TRUE}\) 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], 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
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 start points in the data frame
- lat_start: Column containing the latitudes of the start points in the data frame
- lon_start: Column containing the longitudes of the start points in the data frame
- endpoints: A data frame containing the points to count
- ep_id: Column containing the IDs of the points to count in the data frame
- lat_end: Column containing the latitudes of the points to count in the data frame
- lon_end: Column containing the longitudes of the points to count in the data frame
- ep_sum: Column of an additional variable in the data frame 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: "m" for meters, "km" for kilometers or "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
dist.calc

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
**dist.mat**

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 start points 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 * 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, distNbuf

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)
distNbuf (cities, "citynames", "lat", "lon", cities, "citynames", "lat", "lon", bufdist = 300000)
# Cities within 300 km

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

<table>
<thead>
<tr>
<th>Freiburg</th>
<th>Employment data in Freiburg and Germany</th>
</tr>
</thead>
</table>

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

```r
data(Freiburg)
# Loads the data
growth(Freiburg$Freiburg2008, Freiburg$Freiburg2014, growth.type = "rate")
# Industry-specific growth rates for Freiburg 2008 to 2014
```

---

**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**

```r
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
G.counties.gdp

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
A numeric vector containing the GDP per capita for German counties (Landkreise) for 1999

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2000

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2001

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2002

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2003

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2004

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2005

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2006

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2007

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2008

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2009

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2010

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2011

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2012

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2013

A numeric vector containing the GDP per capita for German counties (Landkreise) for 2014

Region West or East

Details

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

Source

References

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], 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
 gini.conc(finance2008[finance2008$industry==
  "Erbringung von Finanz- und Vers.leistungen (K)",]$emp,
  finance2008[finance2008$industry=="Insbesamt",]$emp)
finance2014 <- G.regions.emp[(G.regions.emp$industry ==
  "Erbringung von Finanz- und Vers.leistungen (K)" | G.regions.emp$industry == "Insbesamt") & G.regions.emp$year == "2014",]
# delete all-over data for all industries
 gini.conc(finance2014[finance2014$industry==
  "Erbringung von Finanz- und Vers.leistungen (K)",]$emp,
  finance2014[finance2014$industry=="Insbesamt",]$emp)

---

gini  
Gini coefficient

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, 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)

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)
lc logical argument that indicates if the Lorenz curve is plotted additionally (de- 
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 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
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(20,50,10,70)
# 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, lcgx = 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)

# 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)
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", 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, lcgNlabX = 0, lcgNlabY = 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 `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")
The Gini coefficient of spatial industry concentration ($G_i$) is a special spatial modification of the Gini coefficient of inequality (see the function \texttt{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 \texttt{lc = TRUE}).

\textbf{Value}

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

\textbf{Author(s)}

Thomas Wieland

\textbf{References}


\textbf{See Also}

\texttt{gini}, \texttt{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$year == "2008",]
finance2008 <- finance2008[finance2008$region != "Insgesamt",]
# delete all-over data for all industries
gini.conc(finance2008[finance2008$industry ==
"Erbringung von Finanz- und Vers.leistungen (K)",]$emp,
finance2008[finance2008$industry == "Insgesamt",]$emp)
finance2014 <- G.regions.emp[(G.regions.emp$industry ==
"Erbringung von Finanz- und Vers.leistungen (K)" |
G.regions.emp$industry == "Insgesamt") & G.regions.emp$year == "2014",]
# delete all-over data for all industries
gini.conc(finance2014[finance2014$industry ==
"Erbringung von Finanz- und Vers.leistungen (K)",]$emp,
finance2014[finance2014$industry == "Insgesamt",]$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"},}
\texttt{lcy = \texttt{"\% of regarded variable"}, lctitle = \texttt{"Lorenz curve"},}
\texttt{le.col = \texttt{"blue"}, lc.col = \texttt{"black"}, lsize = 1, ltype = \texttt{"solid"},}
\texttt{bg.col = \texttt{"gray95"}, bgrid = \texttt{TRUE}, bgrid.col = \texttt{"white"},}
\texttt{bgrid.size = 2, bgrid.type = \texttt{"solid"}, lcg = FALSE, lcgn = FALSE,}
\texttt{lcg.caption = \texttt{NULL}, lcg.lab.x = 0, lcg.lab.y = 1,}
\texttt{add.lc = \texttt{FALSE}, plot.lc = \texttt{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{lc} if \texttt{lc = TRUE} (plot of Lorenz curve), \texttt{lcx} 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{lcg} 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
lg.c.lab.x if lg = TRUE (displaying the Gini coefficient in the plot), lg.c.lab.x specifies the x coordinate of the label

lg.c.lab.y if lg = TRUE (displaying the Gini coefficient in the plot), lg.c.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(30000,15000,10000,500000,50000)
# 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)
growth

# 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)

---

growth (Growth rates)

Description

This function calculates the growth from two input numeric vectors

Usage

growth(val1, val2, growth.type = "growth")

Arguments

val1 First numeric vector (e.g. employment at time \(t\))
val2 Second numeric vector (e.g. employment at time \(t\)) or data frame for times \(t+1, t+2, t+3, ..., t+n\)
growth.type Type of growth value that has to be calculated (absolute values or growth rate)

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_A_t <- c(400,150,150,400)
nation_A_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 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, output.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)
- **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)
- **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 $\gamma$ 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
- **accnorm**: 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)
- **output.results**: logical argument that indicates if the results are shown (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 is 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(50000, 100000)
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)
# Combing 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")

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 (\( H_{eq} = \frac{1}{n} \)).

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)
hoover returns the non-standardized HHI (0.34)
herf(sales, coefnorm=TRUE)
hoover returns the standardized HHI (0.12)
herf(sales, output = "eq")
hoover 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)
hoover(x, weighting = NULL)

Arguments

x A numeric vector (dataset of regional income, e.g. GDP)
weighting A numeric weighting vector (dataset of regional population). If weighting = 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

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

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, 35.03, 112.66, 56.22, 85.61, 56.81)
# GDP of german regions 2015 (in billion EUR)
pop <- pop <- c(18879618, 12843514, 3520331, 2484826, 671489, 1787408, 6176172,
1612362, 7926599, 17865516, 4852882, 395597, 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

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,
output.shares = FALSE, output.totals = FALSE,
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 attractivity
variable (e.g. sales area)
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. 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 attractivity or distance value must be greater than zero.

Value

A list containing the following objects:

- \( \text{ijmatrix} \): A data frame containing the Huff interaction matrix
- \( \text{totals} \): 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
converse, 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")
huff_shares
# 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",
output.totals = TRUE, outputshares = TRUE)
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

# 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, 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(0,0,0,66,40)
# 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_1l)

Arguments

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

e_1l 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):
Swedish <- c(45000, 15000, 32000, 10000, 30000)
Norwegian <- c(35000, 12000, 30000, 8000, 22000)
Danish <- c(40000, 10000, 25000, 9000, 18000)
Finnish <- 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  

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>locq</td>
<td>Location quotient</td>
</tr>
</tbody>
</table>

Description

Calculating the location quotient

Usage

```r
locq(e_ij, e_j, e_i, e)
```

Arguments

- `e_ij`  
a single numeric value with the employment of industry `i` in region `j`
- `e_j`  
a single numeric value with the overall employment in region `j`
- `e_i`  
a single numeric value with the overall employment in industry `i`
- `e`  
a single numeric value with the overall 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

```r
# Example from Farhauer/Kroell (2013):
locq (1714, 79006, 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",
ltitle = "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

### Parameters

- **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 diagonal 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, lcg = 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 5 income classes
sizeofclass <- c(1000, 1200, 1600, 400, 200, 600)
# size of income classes
lorenz (income, weighting = sizeofclass, lcg = TRUE, lcg = 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, lcg = 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).
mssd

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

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

```r
data1 <- c(10,10,10,20,20,30,30,30)
# 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
portfolio

Usage

portfolio(region1, region2, nation1, nation2, industry.names = NULL, 
 psize, psize.factor = 10, 
 pmx = "Regional growth", pmy = "National growth", pmtile = "Portfolio matrix", 
 pcoll = NULL, leg = FALSE, leg.fsize = 1, leg.x = -max_val, leg.y = -max_val/2, 
 bg.col = "gray95", bgrid = TRUE, bgrid.col = "white", bgrid.size = 2, 
 bgrid.type = "solid", seg.x = 0, seg.y = 0)

Arguments

region1 a numeric vector with \( i \) values containing the employment in \( i \) industries in a 
region at time 1

region2 a numeric vector with \( i \) values containing the employment in \( i \) industries in a 
region at time 2

nation1 a numeric vector with \( i \) values containing the employment in \( i \) industries in the 
whole nation at time 1

nation2 a numeric vector with \( i \) values containing the employment in \( i \) industries in the 
whole nation at time 2

industry.names Industry names

psize Point size in the portfolio matrix plot (mostly the absolute values of employment 
in \( i \) industries in the region at time 2)

psize.factor Enlargement factor for the points in the plot

pmx Name of the X axis in the plot

pmy Name of the Y axis in the plot

pmtile Plot title

pcoll Industry-specific point colors

leg Logical argument that indicates if a legend has to be added to the plot

leg.fsize If leg = TRUE: Font size in the plot legend

leg.x If leg = TRUE: X coordinate of the legend

leg.y If leg = TRUE: Y coordinate of the legend

bg.col Background color

bgrid Logical argument that indicates if a grid has to be added to the plot

bgrid.col If bgrid = TRUE: Color of the grid

bgrid.size If bgrid = TRUE: Size of the grid

bgrid.type If bgrid = TRUE: Type of the grid

seg.x X coordinate of segmentation of the plot

seg.y Y coordinate of segmentation of the plot
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

Author(s)

Thomas Wieland

References


See Also

shift, shiftd, 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, pmx = "Freiburg", pmy = "Deutschland", pmtitle = "Freiburg und BRD", pcol = Freiburg$color, leg = TRUE, leg.fsize = 0.6, bgrid = TRUE, leg.y = -0.17)
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.plotSize = 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 (= 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
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 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
sigma.type Estimating sigma convergence via ANOVA (two years) or trend regression (more
  than two years). Default: sigma.type = "anova"
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 conver-
  gence 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 (coef-
  ficient 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 (de-
  fault: 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.plotLS 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

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 \texttt{rca} function is a wrapper for the functions \texttt{betaconv.ols}, \texttt{betaconv.nls}, \texttt{sigmaconv} and \texttt{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.nlm,betaconv.speed,sigmaconv,sigmaconv.t.cv,sd2.var2

Examples

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

rca (G.counties.gdp$gdppc2010, 2010, G.counties.gdp$gdppc2011, 2011,
conditions = NULL, beta.plot = TRUE)
# Two years, no conditions (Absolute beta convergence)
reillypha,P_b, D_a, D_b, gamma = 1, lambda = 2, relation = NULL)

Arguments

- **P_a**: a single numeric value of attractiveness/population size of location/city a
- **P_b**: a single numeric value of attractiveness/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)

---

**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)
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 attractiveness/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:

\[ \text{relation}_{AB} \] relation of trade between cities/locations \( a \) and \( b \)
\[ \text{prop}_A \] proportion of city/location \( a \)
\[ \text{prop}_B \] 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**: 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
plot(years, sd_gdppc, "l", ylim = c(0,15000), xlab = "Year",
ylab = "SD of GDP per capita")
# Plot sd over time
```
Description
Analyzing regional growth with the shift-share analysis

Usage
shift(region1, region2, nation1, nation2, industry.names = NULL,
shift.method = "Dunn", output.results = TRUE, plot.results = FALSE,
plot.colours = NULL, plot.title = NULL, plot.portfolio = FALSE, ...)

Arguments
- region1: a numeric vector with i values containing the employment in i industries in a region at time 1
- region2: a numeric vector with i values containing the employment in i industries in a region at time 2
- nation1: a numeric vector with i values containing the employment in i industries in the whole nation at time 1
- nation2: a numeric vector with i values containing the employment in i industries in the whole nation at time 2
- industry.names: Industry names
- shift.method: Method of shift-share-analysis to be used ("Dunn", "Esteban", "Gerfin") (default: shift.method = "Dunn")
- output.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 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:

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

**Author(s)**

Thomas Wieland

**References**


**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(region1, region2, nation1, nation2, industry.names = NULL)

Arguments

region1 a numeric vector with i values containing the employment in i industries in a region at time 1
region2 a numeric vector with i values containing the employment in i industries in a region at time 2
nation1 a numeric vector with i values containing the employment in i industries in the whole nation at time 1
nation2  a numeric vector with i values containing the employment in i industries in the whole nation at time 2

industry.names  Industry names

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**

```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)
shift.growth(region_A_t, region_A_t1, nation_X_t, nation_X_t1)
```

---

**Description**

Dynamic shift-share analysis

**Usage**

```r
shiftd(region1, region2, nation1, nation2, time1, time2,
industry.names = NULL, shift.method = "Dunn",
gerfin.shifts = "sum", output.results = TRUE,
plot.results = FALSE, plot.colours = NULL, plot.title = NULL,
plot.portfolio = FALSE, ...)```

**Arguments**

- `region1`: a numeric vector with i values containing the employment in i industries in a region at time 1
- `region2`: a data frame with i rows containing the employment in i industries in a region for j years
- `nation1`: a numeric vector with i values containing the employment in i industries in the whole nation at time 1
- `nation2`: a data frame with i rows containing the employment in i industries in the whole nation for j years
- `time1`: Initial year
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)
Usage

shifti(region1, region2, nation1, nation2, industry.names = NULL,
shift.method = "Dunn", output.results = TRUE, plot.results = FALSE,
plot.colours = NULL, plot.title = NULL, plot.portfolio = FALSE, ...)

Arguments

region1 a numeric vector with $i$ values containing the employment in $i$ industries in a region at time 1
region2 a numeric vector with $i$ values containing the employment in $i$ industries in a region at time 2
nation1 a numeric vector with $i$ values containing the employment in $i$ industries in the whole nation at time 1
nation2 a numeric vector with $i$ values containing the employment in $i$ industries in the whole nation at time 2
industry.names Industry names
shift.method Method of shift-share-analysis to be used ("Dunn", "Gerfin") (default: shift.method = "Dunn")
output.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)
shifti(region_A_t, region_A_t1, nation_X_t, nation_X_t1, plot.results = TRUE, plot.portfolio = TRUE, psize = region_A_t1)
sigmaconv

Analysis of regional sigma convergence for two years using ANOVA

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, output.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 = "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.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 = FALSE, so the denominator of variance is \( n \))
output.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, $y_i$ 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 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**


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

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.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", output.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 $t+1, t+2, t+3, ..., t+n$
time2 A single value of time $t+1$

See Also

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

Examples

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
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.plotLineColor 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

output.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


theil

See Also

rca, sigmavec, betaconv, betaconv.speed, cv, sd, var2

Examples

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

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

sigmaconv.t (G.countries.gdp$gdppc2010, 2010, G.countries.gdp[65:68], 2014, 
sigma.measure = "cv", sigma.log = FALSE, output.results = TRUE)
# Using the coefficient of variation (GDP per capita not logged)

theil

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 Theil inequality index (0 < TI < 1).

Author(s)

Thomas Wieland

References

See Also

`gini, herf, hoover`

Examples

```r
# 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

```
var2(x, is.sample = TRUE, weighting = NULL, wmean = FALSE, na.rm = FALSE)
```

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 `variance` (var). Unlike the R base `var` function, the `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 `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 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 (σ_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 variance (optionally using a weighted arithmetic mean if \texttt{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

\texttt{sd2, cv, gini, herf, hoover, mean2, rca}

Examples

\begin{verbatim}
# 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
\end{verbatim}
Index

betaconv.nls, 4, 9, 11, 60, 76, 79
betaconv.ols, 6, 6, 11, 60
betaconv.speed, 6, 9, 10, 60, 76, 79
converse, 11, 36, 41, 62
cv, 6, 9, 11, 13, 27, 37, 39, 51, 54, 60, 76, 79, 82
dist, 16, 18
dist.buf, 15, 17, 18, 36
dist.calc, 16, 36
dist.mat, 16, 17, 17, 36
EU28.emp, 18
Freiburg, 19
G.counties.gdp, 20
G.regions.emp, 23
gini, 14, 24, 29, 32, 37, 39, 64, 80, 82
gini.conc, 27, 28, 32, 43–46, 49, 51
gini.spec, 27, 29, 30, 43–46, 49, 51
growth, 33
hansen, 34, 41
herf, 14, 27, 36, 39, 51, 64, 80, 82
hoover, 14, 27, 38, 51, 64, 80, 82
huff, 12, 36, 39, 62
krugman.conc, 42, 44–46
krugman.conc2, 43, 43, 45, 46
krugman.spec, 43, 44, 44, 46
krugman.spec2, 43–45, 46
lm.beta, 47
locq, 43–46, 48
lorenz, 49
mean2, 52, 64, 82
mssd, 53
portfolio, 54, 66, 69, 71, 73
rca, 6, 9, 14, 57, 64, 76, 79, 82
REAT (REAT-package), 2
REAT-package, 2
reilly, 12, 36, 41, 61
sd2, 6, 9, 11, 13, 27, 37, 39, 51, 60, 76, 79, 82
shift, 56, 65, 69, 71, 73
shift.growth, 66, 67, 71, 73
shifted, 56, 66, 69, 69
shifti, 56, 66, 69, 71, 71, 73
sigmaconv, 6, 9, 11, 60, 74, 79
sigmaconv.t, 6, 9, 11, 60, 76, 76
theil, 79
to.dummy, 80
var2, 6, 9, 11, 54, 60, 76, 79, 81