Package 'EconGeo'

April 26, 2017

Title Computing Key Indicators of the Spatial Distribution of Economic Activities

Version 1.3

Date 2017

Author Pierre-Alexandre Balland <p.balland@uu.nl>

Maintainer Pierre-Alexandre Balland <p.balland@uu.nl>

Description Functions to compute a series of indices commonly used in economic geography & urban economics to describe the location, distribution, spatial organization, and complexity of economic activities. Most of the functions use matrix calculus and are based on bipartite (incidence) matrices consisting of region - industry pairs.

URL https://github.com/PABalland/EconGeo

Depends R (>= 3.3.1)

Imports Matrix,
reshape

License GPL-2 | GPL-3 [expanded from: GPL]

Encoding UTF-8

LazyData true

RoxygenNote 5.0.1.9000

BugReports http://github.com/PABalland/EconGeo/issues

R topics documented:

`co.occurrence` *Compute the number of co-occurrences between industry pairs from an incidence (industry - event) matrix*

Description

This function computes the number of co-occurrences between industry pairs from an incidence (industry - event) matrix

Usage

```
co.occurrence(mat, diagonal = FALSE, list = FALSE)
```

Arguments

`mat`	An incidence matrix with industries in rows and events in columns
`diagonal`	Logical; shall the values in the diagonal of the co-occurrence matrix be included in the output? Defaults to FALSE (values in the diagonal are set to 0), but can be set to TRUE (values in the diagonal reflects in how many events a single industry can be found)
`list`	Logical; is the input a list? Defaults to FALSE (input = adjacency matrix), but can be set to TRUE if the input is an edge list

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

See Also

`relatedness`, `relatedness.density`

Examples

```
## generate a region - events matrix
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 5)
rownames(mat) <- c ("I1", "I2", "I3", "I4")
colnames(mat) <- c("US1", "US2", "US3", "US4", "US5")

## run the function
co.occurrence (mat)
co.occurrence (mat, diagonal = TRUE)

## generate a regular data frame (list)
list <- get.list (mat)

## run the function
co.occurrence (list, list = TRUE)
co.occurrence (list, list = TRUE, diagonal = TRUE)
```

`diversity` *Compute a simple measure of diversity of regions*

Description

This function computes a simple measure of diversity of regions by counting the number of industries in which a region has a relative comparative advantage (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
diversity(mat, RCA = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
RCA	Logical; should the index of relative comparative advantage (RCA - also refered to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

ubiquity, location.quotient

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
diversity (mat, RCA = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
diversity (mat)
```

---

ease.recombination *Compute the ease of recombination of a given technological class*

Description

This function computes the ease of recombination of a given technological class from technological classes - patents (incidence) matrices

Usage

```
ease.recombination(mat, sparse = FALSE, list = FALSE)
```

Arguments

mat	A bipartite adjacency matrix (can be a sparse matrix)
sparse	Logical; is the input matrix a sparse matrix? Defaults to FALSE, but can be set to TRUE if the input matrix is a sparse matrix

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Fleming, L. and Sorenson, O. (2001) Technology as a complex adaptive system: evidence from patent data, *Research Policy* **30**: 1019-1039

See Also

modular.complexity, TCI, MORt

Examples

```
## generate a technology - patent matrix
set.seed(31)
mat <- matrix(sample(0:1,30,replace=T), ncol = 5)
rownames(mat) <- c ("T1", "T2", "T3", "T4", "T5", "T6")
colnames(mat) <- c ("US1", "US2", "US3", "US4", "US5")

## generate a technology - patent sparse matrix
library (Matrix)
smat <- Matrix(mat,sparse=TRUE)

## run the function
ease.recombination (mat)
ease.recombination (smat, sparse = TRUE)

## generate a regular data frame (list)
list <- get.list (mat)

## run the function
ease.recombination (list, list = TRUE)
```

---

entropy — *Compute the Shannon entropy index from regions - industries matrices*

Description

This function computes the Shannon entropy index from regions - industries matrices from (incidence) regions - industries matrices

Usage

```
entropy(mat)
```

Arguments

mat   An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Shannon, C.E., Weaver, W. (1949) *The Mathematical Theory of Communication*. Univ of Illinois Press.

Frenken, K., Van Oort, F. and Verburg, T. (2007) Related variety, unrelated variety and regional economic growth, *Regional studies* **41** (5): 685-697.

See Also

diversity

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
entropy (mat)
```

---

entry.list   *Generate a data frame of entry events from multiple regions - industries matrices (same matrix composition for the different periods)*

Description

This function generates a data frame of entry events from multiple regions - industries matrices (different matrix compositions are allowed). In this function, the maximum number of periods is limited to 20.

Usage

```
entry.list(mat1, mat2, mat3, mat4, mat5, mat6, mat7, mat8, mat9, mat10, mat11,
  mat12, mat13, mat14, mat15, mat16, mat17, mat18, mat19, mat20)
```

Arguments

`mat1`	An incidence matrix with regions in rows and industries in columns (period 1 - mandatory)
`mat2`	An incidence matrix with regions in rows and industries in columns (period 2 - mandatory)
`mat...`	An incidence matrix with regions in rows and industries in columns (period ... - optional)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

`entry`, `exit`, `exit.list`

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
mat2[3,1] <- 1

## run the function
entry.list (mat1, mat2)

## generate a third region - industry matrix in which cells represent the presence/absence
## of a RCA (period 3)
mat3 <- mat2
mat3[5,2] <- 1

## run the function
entry.list (mat1, mat2, mat3)

## generate a fourth region - industry matrix in which cells represent the presence/absence
## of a RCA (period 4)
mat4 <- mat3
```

```
mat4[5,4] <- 1

## run the function
entry.list (mat1, mat2, mat3, mat4)
```

---

entry.mat *Generate a matrix of entry events from two regions - industries matrices (same matrix composition from two different periods)*

Description

This function generates a matrix of entry events from two regions - industries matrices (different matrix compositions are allowed)

Usage

```
entry.mat(mat1, mat2)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1)
mat2	An incidence matrix with regions in rows and industries in columns (period 2)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

exit, entry.list, exit.list

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
```

```
mat2[3,1] <- 1

## run the function
entry.mat (mat1, mat2)
```

---

exit.list *Generate a data frame of exit events from multiple regions - industries matrices (same matrix composition for the different periods)*

Description

This function generates a data frame of exit events from multiple regions - industries matrices (different matrix compositions are allowed). In this function, the maximum number of periods is limited to 20.

Usage

```
exit.list(mat1, mat2, mat3, mat4, mat5, mat6, mat7, mat8, mat9, mat10, mat11,
  mat12, mat13, mat14, mat15, mat16, mat17, mat18, mat19, mat20)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1 - mandatory)
mat2	An incidence matrix with regions in rows and industries in columns (period 2 - mandatory)
mat...	An incidence matrix with regions in rows and industries in columns (period ... - optional)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

entry, exit, entry.list

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
mat2[2,1] <- 0

## run the function
exit.list (mat1, mat2)

## generate a third region - industry matrix in which cells represent the presence/absence
## of a RCA (period 3)
mat3 <- mat2
mat3[5,1] <- 0

## run the function
exit.list (mat1, mat2, mat3)

## generate a fourth region - industry matrix in which cells represent the presence/absence
## of a RCA (period 4)
mat4 <- mat3
mat4[5,3] <- 0

## run the function
exit.list (mat1, mat2, mat3, mat4)
```

---

exit.mat — *Generate a matrix of exit events from two regions - industries matrices (same matrix composition from two different periods)*

Description

This function generates a matrix of exit events from two regions - industries matrices (different matrix compositions are allowed)

Usage

```
exit.mat(mat1, mat2)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1)
mat2	An incidence matrix with regions in rows and industries in columns (period 2)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Wolf-Hendrik Uhlbach <w.p.uhlbach@students.uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

entry, exit.list, entry.list

Examples

```
## generate a first region - industry matrix in which cells represent the presence/absence
## of a RCA (period 1)
set.seed(31)
mat1 <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix in which cells represent the presence/absence
## of a RCA (period 2)
mat2 <- mat1
mat2[2,1] <- 0


## run the function
exit.mat (mat1, mat2)
```

---

expy — *Compute the expy index of regions from regions - industries matrices*

Description

This function computes the expy index of regions from (incidence) regions - industries matrices, as proposed by Hausmann, Hwang & Rodrik (2007). The index is a measure of the productivity level associated with a region's specialization pattern.

Usage

```
expy(mat, vec)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
vec	A vector that gives GDP, R&D, education or any other relevant regional attribute that will be used to compute the weighted average for each industry

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123

Hausmann, R., Hwang, J. & Rodrik, D. (2007) What you export matters, *Journal of economic growth* **12**: 1-25.

See Also

`location.quotient`

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## a vector of GDP of regions
vec <- c (5, 10, 15, 25, 50)
## run the function
expy (mat, vec)
```

`get.list` *Create regular data frames from regions - industries matrices*

Description

This function creates regular data frames with three columns (regions, industries, count) from (incidence) matrices (wide to long format) using the reshape2 package

Usage

```
get.list (data)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns (or the other way around)
`sparse`	Logical; is the input a sparse matrix? Defaults to FALSE

Author(s)

Pierre-Alexandre Balland <`p.balland@uu.nl`>

See Also

`get.matrix`

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
get.list (mat)
```

`get.matrix` *Create regions - industries matrices from regular data frames*

Description

This function creates regions - industries (incidence) matrices from regular data frames (long to wide format) using the reshape2 package or the Matrix package

Usage

```
get.matrix (data)
```

Arguments

`data`	is a data frame with three columns (regions, industries, count)
`sparse`	Logical; shall the returned output be a sparse matrix? Defaults to FALSE, but can be set to TRUE if the dataset is very large

Author(s)

Pierre-Alexandre Balland `<p.balland@uu.nl>`

See Also

`get.list`

Examples

```
## generate a region - industry data frame
set.seed(31)
region <- c("R1", "R1", "R1", "R1", "R2", "R2", "R3", "R4", "R5", "R5")
industry <- c("I1", "I2", "I3", "I4", "I1", "I2", "I1", "I1", "I3", "I3")
data <- data.frame (region, industry)
data$count <- 1

## run the function
get.matrix (data)
get.matrix (data, sparse = TRUE)
```

Gini *Compute the Gini coefficient*

Description

This function computes the Gini coefficient. The Gini index measures spatial inequality. It ranges from 0 (perfect income equality) to 1 (perfect income inequality) and is derived from the Lorenz curve. The Gini coefficient is defined as a ratio of two surfaces derived from the Lorenz curve. The numerator is given by the area between the Lorenz curve of the distribution and the uniform distribution line (45 degrees line). The denominator is the area under the uniform distribution line (the lower triangle). This index gives an indication of the unequal distribution of an industry accross n regions. Maximum inequality in the sample occurs when n-1 regions have a score of zero and one region has a positive score. The maximum value of the Gini coefficient is (n-1)/n and approaches 1 (theoretical maximum limit) as the number of observations (regions) increases.

Usage

```
Gini(mat)
```

Arguments

`ind`	A vector of industrial regional count

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Gini, C. (1921) Measurement of Inequality of Incomes, *The Economic Journal* **31**: 124-126

See Also

`Hoover.Gini`, `locational.Gini`, `locational.Gini.curve`, `Lorenz.curve`, `Hoover.curve`

Examples

```
## generate vectors of industrial count
ind <- c(0, 10, 10, 30, 50)

## run the function
Gini (ind)

## generate a region - industry matrix
mat = matrix (
c (0, 1, 0, 0,
0, 1, 0, 0,
0, 1, 0, 0,
0, 1, 0, 1,
0, 1, 1, 1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")
```

```
## run the function
Gini (mat)

## run the function by aggregating all industries
Gini (rowSums(mat))

## run the function for industry #1 only (perfect equality)
Gini (mat[,1])

## run the function for industry #2 only (perfect equality)
Gini (mat[,2])

## run the function for industry #3 only (perfect unequality: max Gini = (5-1)/5)
Gini (mat[,3])

## run the function for industry #4 only (top 40% produces 100% of the output)
Gini (mat[,4])
```

`growth.ind` *Generate a matrix of industrial growth by industries from two regions - industries matrices (same matrix composition from two different periods)*

Description

This function generates a matrix of industrial growth by industries from two regions - industries matrices (same matrix composition from two different periods)

Usage

```
growth.ind(mat1, mat2)
```

Arguments

`mat1`	An incidence matrix with regions in rows and industries in columns (period 1)
`mat2`	An incidence matrix with regions in rows and industries in columns (period 2)

Author(s)

Pierre-Alexandre Balland <`p.balland@uu.nl`>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

`exit`, `entry.list`, `exit.list`

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3,1] <- 8


## run the function
growth.ind (mat1, mat2)
```

`growth.list` *Generate a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods)*

Description

This function generates a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods). In this function, the maximum number of periods is limited to 20.

Usage

```
growth.list(mat1, mat2, mat3, mat4, mat5, mat6, mat7, mat8, mat9, mat10, mat11,
  mat12, mat13, mat14, mat15, mat16, mat17, mat18, mat19, mat20)
```

Arguments

`mat1`	An incidence matrix with regions in rows and industries in columns (period 1 - mandatory)
`mat2`	An incidence matrix with regions in rows and industries in columns (period 2 - mandatory)
`mat...`	An incidence matrix with regions in rows and industries in columns (period ... - optional)

Author(s)

Pierre-Alexandre Balland <`p.balland@uu.nl`>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

growth, exit, exit.list

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3,1] <- 8

## run the function
growth.list (mat1, mat2)

## generate a third region - industry matrix with full count (period 3)
mat3 <- mat2
mat3[5,2] <- 1

## run the function
growth.list (mat1, mat2, mat3)

## generate a fourth region - industry matrix with full count (period 4)
mat4 <- mat3
mat4[5,4] <- 1

## run the function
growth.list (mat1, mat2, mat3, mat4)
```

---

growth.list.ind — *Generate a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods)*

Description

This function generates a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods). In this function, the maximum number of periods is limited to 20.

Usage

```
growth.list.ind(mat1, mat2, mat3, mat4, mat5, mat6, mat7, mat8, mat9, mat10,
  mat11, mat12, mat13, mat14, mat15, mat16, mat17, mat18, mat19, mat20)
```

Arguments

mat1	An incidence matrix with regions in rows and industries in columns (period 1 - mandatory)

`mat2`	An incidence matrix with regions in rows and industries in columns (period 2 - mandatory)
`mat...`	An incidence matrix with regions in rows and industries in columns (period ... - optional)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

growth, exit, exit.list

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3,1] <- 8

## run the function
growth.list.ind (mat1, mat2)

## generate a third region - industry matrix with full count (period 3)
mat3 <- mat2
mat3[5,2] <- 1

## run the function
growth.list.ind (mat1, mat2, mat3)

## generate a fourth region - industry matrix with full count (period 4)
mat4 <- mat3
mat4[5,4] <- 1

## run the function
growth.list.ind (mat1, mat2, mat3, mat4)
```

growth.list.reg — *Generate a data frame of region growth from multiple regions - industries matrices (same matrix composition for the different periods)*

Description

This function generates a data frame of industrial growth in regions from multiple regions - industries matrices (same matrix composition for the different periods). In this function, the maximum number of periods is limited to 20.

Usage

```
growth.list.reg(mat1, mat2, mat3, mat4, mat5, mat6, mat7, mat8, mat9, mat10,
  mat11, mat12, mat13, mat14, mat15, mat16, mat17, mat18, mat19, mat20)
```

Arguments

`mat1`	An incidence matrix with regions in rows and industries in columns (period 1 - mandatory)
`mat2`	An incidence matrix with regions in rows and industries in columns (period 2 - mandatory)
`mat...`	An incidence matrix with regions in rows and industries in columns (period ... - optional)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

growth, exit, exit.list

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3,1] <- 8
```

```
## run the function
growth.list.reg (mat1, mat2)

## generate a third region - industry matrix with full count (period 3)
mat3 <- mat2
mat3[5,2] <- 1

## run the function
growth.list.reg (mat1, mat2, mat3)

## generate a fourth region - industry matrix with full count (period 4)
mat4 <- mat3
mat4[5,4] <- 1

## run the function
growth.list.reg (mat1, mat2, mat3, mat4)
```

`growth.mat` *Generate a matrix of industrial growth in regions from two regions - industries matrices (same matrix composition from two different periods)*

Description

This function generates a matrix of industrial growth in regions from two regions - industries matrices (same matrix composition from two different periods)

Usage

```
growth.mat(mat1, mat2)
```

Arguments

`mat1`	An incidence matrix with regions in rows and industries in columns (period 1)
`mat2`	An incidence matrix with regions in rows and industries in columns (period 2)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

`exit`, `entry.list`, `exit.list`

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3,1] <- 8


## run the function
growth.mat (mat1, mat2)
```

---

growth.reg — *Generate a matrix of industrial growth by regions from two regions - industries matrices (same matrix composition from two different periods)*

Description

This function generates a matrix of industrial growth by regions from two regions - industries matrices (same matrix composition from two different periods)

Usage

```
growth.reg(mat1, mat2)
```

Arguments

`mat1`	An incidence matrix with regions in rows and industries in columns (period 1)
`mat2`	An incidence matrix with regions in rows and industries in columns (period 2)

Author(s)

Pierre-Alexandre Balland <`p.balland@uu.nl`>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

`exit`, `entry.list`, `exit.list`

Examples

```
## generate a first region - industry matrix with full count (period 1)
set.seed(31)
mat1 <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix with full count (period 2)
mat2 <- mat1
mat2[3,1] <- 8


## run the function
growth.reg (mat1, mat2)
```

Hachman — *Compute the Hachman index from regions - industries matrices*

Description

This function computes the Hachman index from regions - industries matrices. The Hachman index indicates how closely the industrial distribution of a region resembles the one of a more global economy (nation, world). The index varies between 0 (extreme dissimilarity between the region and the more global economy) and 1 (extreme similarity between the region and the more global economy)

Usage

```
Hachman(mat)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

See Also

average.location.quotient

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
Hachman (mat)
```

`Herfindahl` *Compute the Herfindahl index from regions - industries matrices*

Description

This function computes the Herfindahl index from regions - industries matrices from (incidence) regions - industries matrices. This index is also known as the Herfindahl-Hirschman index.

Usage

```
Herfindahl(mat)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland `<p.balland@uu.nl>`

References

Herfindahl, O.C. (1959) *Copper Costs and Prices: 1870-1957.* Baltimore: The Johns Hopkins Press.

Hirschman, A.O. (1945) *National Power and the Structure of Foreign Trade*, Berkeley and Los Angeles: University of California Press.

See Also

`Krugman.index`

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
Herfindahl (mat)
```

Hoover.curve — *Plot a Hoover curve from regions - industries matrices*

Description

This function plots a Hoover curve from regions - industries matrices.

Usage

```
Hoover.curve(mat, pop, plot = TRUE, pdf = FALSE)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
`pop`	A vector of population regional count
`plot`	Logical; shall the curve be automatically plotted? Defaults to TRUE. If set to TRUE, the function will return x y coordinates that you can latter use to plot and customize the curve.
`pdf`	Logical; shall a pdf be saved to your current working directory? Defaults to FALSE. If set to TRUE, a pdf with all Hoover curves will be compiled and saved to your current working directory.

Author(s)

Pierre-Alexandre Balland `<p.balland@uu.nl>`

References

Hoover, E.M. (1936) The Measurement of Industrial Localization, *The Review of Economics and Statistics* **18** (1): 162-171

See Also

`Hoover.Gini`, `locational.Gini`, `locational.Gini.curve`, `Lorenz.curve`, `Gini`

Examples

```
## generate vectors of industrial and population count
ind <- c(0, 10, 10, 30, 50)
pop <- c(10, 15, 20, 25, 30)

## run the function (30% of the population produces 50% of the industrial output)
Hoover.curve (ind, pop)
Hoover.curve (ind, pop, pdf = TRUE)
Hoover.curve (ind, pop, plot = F)

## generate a region - industry matrix
mat = matrix (
c (0, 10, 0, 0,
0, 15, 0, 0,
```

```
0, 20, 0, 0,
0, 25, 0, 1,
0, 30, 1, 1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
Hoover.curve (mat, pop)
Hoover.curve (mat, pop, pdf = TRUE)
Hoover.curve (mat, pop, plot = FALSE)

## run the function by aggregating all industries
Hoover.curve (rowSums(mat), pop)
Hoover.curve (rowSums(mat), pop, pdf = TRUE)
Hoover.curve (rowSums(mat), pop, plot = FALSE)

## run the function for industry #1 only
Hoover.curve (mat[,1], pop)
Hoover.curve (mat[,1], pop, pdf = TRUE)
Hoover.curve (mat[,1], pop, plot = FALSE)

## run the function for industry #2 only (perfectly proportional to population)
Hoover.curve (mat[,2], pop)
Hoover.curve (mat[,2], pop, pdf = TRUE)
Hoover.curve (mat[,2], pop, plot = FALSE)

## run the function for industry #3 only (30% of the pop. produces 100% of the output)
Hoover.curve (mat[,3], pop)
Hoover.curve (mat[,3], pop, pdf = TRUE)
Hoover.curve (mat[,3], pop, plot = FALSE)

## run the function for industry #4 only (55% of the pop. produces 100% of the output)
Hoover.curve (mat[,4], pop)
Hoover.curve (mat[,4], pop, pdf = TRUE)
Hoover.curve (mat[,4], pop, plot = FALSE)

Compare the distribution of the #industries
par(mfrow=c(2,2))
Hoover.curve (mat[,1], pop)
Hoover.curve (mat[,2], pop)
Hoover.curve (mat[,3], pop)
Hoover.curve (mat[,4], pop)
```

`Hoover.Gini` *Compute the Hoover Gini*

Description

This function computes the Hoover Gini, named after Hedgar Hoover. The Hoover index is a measure of spatial inequality. It ranges from 0 (perfect equality) to 1 (perfect inequality) and is calculated from the Hoover curve associated with a given distribution of population, industries or technologies and a reference category. In this sense, it is closely related to the Gini coefficient and the Hoover index. The numerator is given by the area between the Hoover curve of the distribution and the uniform distribution line (45 degrees line). The denominator is the area under the uniform distribution line (the lower triangle).

Usage

```
Hoover.Gini(mat, pop)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
`pop`	A vector of population regional count

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. (1936) The Measurement of Industrial Localization, *The Review of Economics and Statistics* **18** (1): 162-171

See Also

`Hoover.curve`, `locational.Gini`, `locational.Gini.curve`, `Lorenz.curve`, `Gini`

Examples

```
## generate vectors of industrial and population count
ind <- c(0, 10, 10, 30, 50)
pop <- c(10, 15, 20, 25, 30)

## run the function (30% of the population produces 50% of the industrial output)
Hoover.Gini (ind, pop)

## generate a region - industry matrix
mat = matrix (
c (0, 10, 0, 0,
0, 15, 0, 0,
0, 20, 0, 0,
0, 25, 0, 1,
0, 30, 1, 1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
Hoover.Gini (mat, pop)

## run the function by aggregating all industries
Hoover.Gini (rowSums(mat), pop)

## run the function for industry #1 only
Hoover.Gini (mat[,1], pop)

## run the function for industry #2 only (perfectly proportional to population)
```

```
Hoover.Gini (mat[,2], pop)

## run the function for industry #3 only (30% of the pop. produces 100% of the output)
Hoover.Gini (mat[,3], pop)

## run the function for industry #4 only (55% of the pop. produces 100% of the output)
Hoover.Gini (mat[,4], pop)
```

`Hoover.index` *Compute the Hoover index*

Description

This function computes the Hoover index, named after Hedgar Hoover. The Hoover index is a measure of spatial inequality. It ranges from 0 (perfect equality) to 100 (perfect inequality) and is calculated from the Lorenz curve associated with a given distribution of population, industries or technologies. In this sense, it is closely related to the Gini coefficient. The Hoover index represents the maximum vertical distance between the Lorenz curve and the 45 degree line of perfect spatial equality. It indicates the proportion of industries, jobs, or population needed to be transferred from the top to the bottom of the distribution to achieve perfect spatial equality. The Hoover index is also known as the Robin Hood index in studies of income inequality.

Computation of the Hoover index: $H = 1/2 \sum_{i=1}^{N} \left| \frac{E_i}{E_{total}} - \frac{A_i}{A_{total}} \right|$

Usage

```
Hoover.index(mat, pop)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
`pop`	A vector of population regional count; if this argument is missing an equal distribution of the reference group will be assumed.
`pdf`	Logical; shall a pdf be saved to your current working directory? Defaults to FALSE. If set to TRUE, a pdf with all Hoover indices will be compiled and saved to your current working directory.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. (1936) The Measurement of Industrial Localization, *The Review of Economics and Statistics* **18** (1): 162-171

See Also

`Hoover.curve`, `Hoover.Gini`, `locational.Gini`, `locational.Gini.curve`, `Lorenz.curve`, `Gini`

Examples

```
## generate vectors of industrial and population count
ind <- c(0, 10, 10, 30, 50)
pop <- c(10, 15, 20, 25, 30)

## run the function (30% of the population produces 50% of the industrial output)
Hoover.index (ind, pop)

## generate a region - industry matrix
mat = matrix (
c (0, 10, 0, 0,
0, 15, 0, 0,
0, 20, 0, 0,
0, 25, 0, 1,
0, 30, 1, 1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
Hoover.index (mat, pop)

## run the function by aggregating all industries
Hoover.index (rowSums(mat), pop)

## run the function for industry #1 only
Hoover.index (mat[,1], pop)

## run the function for industry #2 only (perfectly proportional to population)
Hoover.index (mat[,2], pop)

## run the function for industry #3 only (30% of the pop. produces 100% of the output)
Hoover.index (mat[,3], pop)

## run the function for industry #4 only (55% of the pop. produces 100% of the output)
Hoover.index (mat[,4], pop)
```

`inv.norm.ubiquity` *Compute a measure of complexity from the inverse of the normalized ubiquity of industries*

Description

This function computes a measure of complexity from the inverse of the normalized ubiquity of industries. We divide the logarithm of the total count (employment, number of firms, number of patents, ...) in an industry by its ubiquity. Ubiquity is given by the number of regions in which an industry can be found (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
inv.norm.ubiquity(mat)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland `<p.balland@uu.nl>`

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

`diversity`, `location.quotient`, `ubiquity`, `TCI`, `MORt`

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
inv.norm.ubiquity (mat)
```

KCI — *Compute an index of knowledge complexity of regions using the eigenvector method*

Description

This function computes an index of knowledge complexity of regions using the eigenvector method from regions - industries (incidence) matrices. Technically, the function returns the eigenvector associated with the second largest eigenvalue of the projected region - region matrix.

Usage

```
KCI(mat, RCA = FALSE)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns
`RCA`	Logical; should the index of relative comparative advantage (RCA - also refered to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Author(s)

Pierre-Alexandre Balland `<p.balland@uu.nl>`

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

location.quotient, ubiquity, diversity, MORc, TCI, MORt

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
KCI (mat, RCA = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
KCI (mat)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
products <- c("P1","P2", "P3", "P4", "P2", "P3", "P4", "P4")
data <- data.frame(countries, products)
data$freq <- 1
mat <- get.matrix (data)

## run the function
KCI (mat)
```

Krugman.index — *Compute the Krugman index from regions - industries matrices*

Description

This function computes the Krugman index from regions - industries matrices. The higher the coefficient, the greater the regional specialization. This index is often referred to as the Krugman specialisation index and measures the distance between the distributions of industry shares in a region and at a more aggregated level (country for instance).

Usage

```
Krugman.index(mat)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Krugman P. (1991) *Geography and Trade*, MIT Press, Cambridge

See Also

average.location.quotient

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
Krugman.index (mat)
```

---

location.quotient *Compute location quotients from regions - industries matrices*

Description

This function computes location quotients from (incidence) regions - industries matrices. The numerator is the share of a given industry in a given region. The denominator is the share of a this industry in a larger economy (overall country for instance). This index is also refered to as the index of Revealed Comparative Advantage (RCA) following Ballasa (1965), or the Hoover-Balassa index.

Usage

```
location.quotient(mat, binary = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
binary	Logical; shall the returned output be a dichotomized version (0/1) of the location quotient? Defaults to FALSE (the full values of the location quotient will be returned), but can be set to TRUE (location quotient values above 1 will be set to 1 & location quotient values below 1 will be set to 0)

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123.

See Also

RCA

Examples

```
## generate a region - industry matrix
mat = matrix (
c (100, 0, 0, 0, 0,
0, 15, 5, 70, 10,
0, 20, 10, 20, 50,
0, 25, 30, 5, 40,
0, 40, 55, 5, 0), ncol = 5, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4", "I5")

## run the function
location.quotient (mat)
location.quotient (mat, binary = TRUE)
```

---

location.quotient.avg — *Compute average location quotients of regions from regions - industries matrices*

Description

This function computes the average location quotients of regions from (incidence) regions - industries matrices. This index is also referred to as the *coefficient of specialization* (Hoover and Giarratani, 1985).

Usage

```
location.quotient.avg(mat)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. and Giarratani, F. (1985) *An Introduction to Regional Economics*. 3rd edition. New York: Alfred A. Knopf

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

See Also

location.quotient, Hachman

Examples

```
## generate a region - industry matrix
mat = matrix (
c (100, 0, 0, 0, 0,
0, 15, 5, 70, 10,
0, 20, 10, 20, 50,
0, 25, 30, 5, 40,
0, 40, 55, 5, 0), ncol = 5, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4", "I5")

## run the function
location.quotient.avg (mat)
```

`locational.Gini` *Compute the locational Gini coefficient from regions - industries matrices*

Description

This function computes the locational Gini coefficient as proposed by Krugman from regions - industries matrices. The higher the coefficient (theoretical limit = 0.5), the greater the industrial concentration. The locational Gini of an industry that is not localized at all (perfectly spread out) in proportion to overall employment would be 0.

Usage

```
locational.Gini(mat)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Krugman P. (1991) *Geography and Trade*, MIT Press, Cambridge (chapter 2 - p.56)

See Also

Hoover.Gini, locational.Gini.curve, Hoover.curve, Lorenz.curve, Gini

Examples

```
## generate a region - industry matrix
mat = matrix (
c (100, 0, 0, 0, 0,
0, 15, 5, 70, 10,
0, 20, 10, 20, 50,
0, 25, 30, 5, 40,
0, 40, 55, 5, 0), ncol = 5, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4", "I5")

## run the function
locational.Gini (mat)
```

---

locational.Gini.curve *Plot a locational Gini curve from regions - industries matrices*

Description

This function plots a locational Gini curve following Krugman from regions - industries matrices.

Usage

```
locational.Gini.curve(mat, pdf = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
pdf	Logical; shall a pdf be saved to your current working directory? Defaults to FALSE. If set to TRUE, a pdf with all locational Gini curves will be compiled and saved to your current working directory.
pop	A vector of population regional count

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Krugman P. (1991) *Geography and Trade*, MIT Press, Cambridge (chapter 2 - p.56)

See Also

Hoover.Gini, locational.Gini, Hoover.curve, Lorenz.curve, Gini

Examples

```
## generate a region - industry matrix
mat = matrix (
c (100, 0, 0, 0, 0,
0, 15, 5, 70, 10,
0, 20, 10, 20, 50,
0, 25, 30, 5, 40,
0, 40, 55, 5, 0), ncol = 5, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4", "I5")

## run the function (shows industry #5)
locational.Gini.curve (mat)
locational.Gini.curve (mat, pdf = TRUE)
```

Lorenz.curve — *Plot a Lorenz curve from regional industrial counts*

Description

This function plots a Lorenz curve from regional industrial counts. This curve gives an indication of the unequal distribution of an industry accross regions.

Usage

```
Lorenz.curve(mat, pdf = FALSE, plot = TRUE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns. The input can also be a vector of industrial regional count (a matrix with n regions in rows and a single column).
pdf	Logical; shall a pdf be saved to your current working directory? Defaults to FALSE. If set to TRUE, a pdf with all Lorenz curves will be compiled and saved to your current working directory.
plot	Logical; shall the curve be automatically plotted? Defaults to TRUE. If set to TRUE, the function will return x y coordinates that you can latter use to plot and customize the curve.

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Lorenz, M. O. (1905) Methods of measuring the concentration of wealth, *Publications of the American Statistical Association* **9**: 209–219

See Also

Hoover.Gini, locational.Gini, locational.Gini.curve, Hoover.curve, Gini

Examples

```
## generate vectors of industrial count
ind <- c(0, 10, 10, 30, 50)

## run the function
Lorenz.curve (ind)
Lorenz.curve (ind, pdf = TRUE)
Lorenz.curve (ind, plot = FALSE)

## generate a region - industry matrix
mat = matrix (
c (0, 1, 0, 0,
0, 1, 0, 0,
0, 1, 0, 0,
0, 1, 0, 1,
0, 1, 1, 1), ncol = 4, byrow = T)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
Lorenz.curve (mat)
Lorenz.curve (mat, pdf = TRUE)
Lorenz.curve (mat, plot = FALSE)

## run the function by aggregating all industries
Lorenz.curve (rowSums(mat))
Lorenz.curve (rowSums(mat), pdf = TRUE)
Lorenz.curve (rowSums(mat), plot = FALSE)

## run the function for industry #1 only (perfect equality)
Lorenz.curve (mat[,1])
Lorenz.curve (mat[,1], pdf = TRUE)
Lorenz.curve (mat[,1], plot = FALSE)

## run the function for industry #2 only (perfect equality)
Lorenz.curve (mat[,2])
Lorenz.curve (mat[,2], pdf = TRUE)
Lorenz.curve (mat[,2], plot = FALSE)

## run the function for industry #3 only (perfect unequality)
Lorenz.curve (mat[,3])
Lorenz.curve (mat[,3], pdf = TRUE)
Lorenz.curve (mat[,3], plot = FALSE)

## run the function for industry #4 only (top 40% produces 100% of the output)
Lorenz.curve (mat[,4])
Lorenz.curve (mat[,4], pdf = TRUE)
Lorenz.curve (mat[,4], plot = FALSE)

Compare the distribution of the #industries
par(mfrow=c(2,2))
Lorenz.curve (mat[,1])
Lorenz.curve (mat[,2])
Lorenz.curve (mat[,3])
Lorenz.curve (mat[,4])
```

`match.mat` *Re-arrange the dimension of a matrix based on the dimension of another matrix*

Description

This function e-arranges the dimension of a matrix based on the dimension of another matrix

Usage

```
match.mat(fill = mat1, dim = mat2, missing = T)
```

Arguments

`fill`	A matrix that will be used to populate the matrix output
`dim`	A matrix that will be used to determine the dimensions of the matrix output
`missing`	Logical; Shall the cells of the non matching rows/columns set to NA? Default to TRUE but can be set to FALSE to set the cells of the non matching rows/columns to 0 instead.

Author(s)

Pierre-Alexandre Balland `<p.balland@uu.nl>`

See Also

`location.quotient`

Examples

```
## generate a first region - industry matrix
set.seed(31)
mat1 <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat1) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat1) <- c ("I1", "I2", "I3", "I4")

## generate a second region - industry matrix
set.seed(31)
mat2 <- matrix(sample(0:1,16,replace=T), ncol = 4)
rownames(mat2) <- c ("R1", "R2", "R3", "R5")
colnames(mat2) <- c ("I1", "I2", "I3", "I4")

## run the function
match.mat (fill = mat1, dim = mat2)
match.mat (fill = mat2, dim = mat1)
match.mat (fill = mat2, dim = mat1, missing = F)
```

modular.complexity *Compute a measure of modular complexity of patent documents*

Description

This function computes a measure of modular complexity of patent documents from technological classes - patents (incidence) matrices

Usage

```
modular.complexity(mat, sparse = FALSE, list = FALSE)
```

Arguments

mat	A bipartite adjacency matrix (can be a sparse matrix)
sparse	Logical; is the input matrix a sparse matrix? Defaults to FALSE, but can be set to TRUE if the input matrix is a sparse matrix
list	Logical; is the input a list? Defaults to FALSE (input = adjacency matrix), but can be set to TRUE if the input is an edge list

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Fleming, L. and Sorenson, O. (2001) Technology as a complex adaptive system: evidence from patent data, *Research Policy* **30**: 1019-1039

See Also

ease.recombination, TCI, MORt

Examples

```
## generate a technology - patent matrix
set.seed(31)
mat <- matrix(sample(0:1,30,replace=T), ncol = 5)
rownames(mat) <- c ("T1", "T2", "T3", "T4", "T5", "T6")
colnames(mat) <- c ("US1", "US2", "US3", "US4", "US5")

## run the function
modular.complexity (mat)

## generate a technology - patent sparse matrix
library (Matrix)

## run the function
smat <- Matrix(mat,sparse=TRUE)

modular.complexity (smat, sparse = TRUE)
## generate a regular data frame (list)
list <- get.list (mat)
```

```
## run the function
modular.complexity (list, list = TRUE)
```

---

modular.complexity.avg *Compute a measure of average modular complexity of technologies*

Description

This function computes a measure of average modular complexity of technologies (average complexity of patent documents in a given technological class) from technological classes - patents (incidence) matrices

Usage

```
modular.complexity.avg(mat, sparse = FALSE, list = FALSE)
```

Arguments

mat	A bipartite adjacency matrix (can be a sparse matrix)
sparse	Logical; is the input matrix a sparse matrix? Defaults to FALSE, but can be set to TRUE if the input matrix is a sparse matrix
list	Logical; is the input a list? Defaults to FALSE (input = adjacency matrix), but can be set to TRUE if the input is an edge list

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Fleming, L. and Sorenson, O. (2001) Technology as a complex adaptive system: evidence from patent data, *Research Policy* **30**: 1019-1039

See Also

ease.recombination, TCI, MORt

Examples

```
## generate a technology - patent matrix
set.seed(31)
mat <- matrix(sample(0:1,30,replace=T), ncol = 5)
rownames(mat) <- c ("T1", "T2", "T3", "T4", "T5", "T6")
colnames(mat) <- c ("US1", "US2", "US3", "US4", "US5")

## run the function
modular.complexity.avg (mat)

## generate a technology - patent sparse matrix
library (Matrix)
```

```
## run the function
smat <- Matrix(mat,sparse=TRUE)

modular.complexity.avg (smat, sparse = TRUE)
## generate a regular data frame (list)
list <- get.list (mat)

## run the function
modular.complexity.avg (list, list = TRUE)
```

MORc *Compute an index of knowledge complexity of regions using the method of reflection*

Description

This function computes an index of knowledge complexity of regions using the method of reflection from regions - industries (incidence) matrices. The index has been developed by Hidalgo and Hausmann (2009) for country - product matrices and adapted by Balland and Rigby (2016) to city - technology matrices.

Usage

```
MORc(mat, RCA = FALSE, steps = 20)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
RCA	Logical; should the index of relative comparative advantage (RCA - also refered to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed
steps	Number of iteration steps. Defaults to 20, but can be set to 0 to give diversity (number of industry in which a region has a RCA), to 1 to give the average ubiquity of the industries in which a region has a RCA, to 2 to give the average diversity of regions that have similar industrial structures, or to any other number of steps < or = to 22. Note that above steps = 2 the index will be rescaled from 0 (minimum relative complexity) to 100 (maximum relative complexity).

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

location.quotient, ubiquity, diversity, KCI, TCI, MORt

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
MORc (mat, RCA = TRUE)
MORc (mat, RCA = TRUE, steps = 0)
MORc (mat, RCA = TRUE, steps = 1)
MORc (mat, RCA = TRUE, steps = 2)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(32)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
MORc (mat)
MORc (mat, steps = 0)
MORc (mat, steps = 1)
MORc (mat, steps = 2)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
products <- c("P1","P2", "P3", "P4", "P2", "P3", "P4", "P4")
data <- data.frame(countries, products)
data$freq <- 1
mat <- get.matrix (data)

## run the function
MORc (mat)
MORc (mat, steps = 0)
MORc (mat, steps = 1)
MORc (mat, steps = 2)
```

MORt — *Compute an index of knowledge complexity of industries using the method of reflection*

Description

This function computes an index of knowledge complexity of industries using the method of reflection from regions - industries (incidence) matrices. The index has been developed by Hidalgo and Hausmann (2009) for country - product matrices and adapted by Balland and Rigby (2016) to city - technology matrices.

Usage

```
MORt(mat, RCA = FALSE, steps = 19)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
RCA	Logical; should the index of relative comparative advantage (RCA - also refered to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed
steps	Number of iteration steps. Defaults to 19, but can be set to 0 to give ubiquity (number of regions that have a RCA in a industry), to 1 to give the average diversity of the regions that have a RCA in this industry, to 2 to give the average ubiquity of technologies developed in the same regions, or to any other number of steps < or = to 21. Note that above steps = 2 the index will be rescaled from 0 (minimum relative complexity) to 100 (maximum relative complexity).

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

location.quotient, ubiquity, diversity, KCI, TCI, MORc

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
MORt (mat, RCA = TRUE)
MORt (mat, RCA = TRUE, steps = 0)
MORt (mat, RCA = TRUE, steps = 1)
MORt (mat, RCA = TRUE, steps = 2)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(32)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
```

```
MORt (mat)
MORt (mat, steps = 0)
MORt (mat, steps = 1)
MORt (mat, steps = 2)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
products <- c("P1","P2", "P3", "P4", "P2", "P3", "P4", "P4")
data <- data.frame(countries, products)
data$freq <- 1
mat <- get.matrix (data)

## run the function
MORt (mat)
MORt (mat, steps = 0)
MORt (mat, steps = 1)
MORt (mat, steps = 2)
```

`norm.ubiquity` *Compute a measure of complexity by normalizing ubiquity of industries*

Description

This function computes a measure of complexity by normalizing ubiquity of industries. We divide the share of the total count (employment, number of firms, number of patents, ...) in an industry by its share of ubiquity. Ubiquity is given by the number of regions in which an industry can be found (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
norm.ubiquity(mat)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland <`p.balland@uu.nl`>

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

`diversity`, `location.quotient`, `ubiquity`, `TCI`, `MORt`

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
norm.ubiquity (mat)
```

`prody` *Compute the prody index of industries from regions - industries matrices*

Description

This function computes the prody index of industries from (incidence) regions - industries matrices, as proposed by Hausmann, Hwang & Rodrik (2007). The index gives an associated income level for each industry. It represents a weighted average of per-capita GDPs (but GDP can be replaced by R&D, education...), where the weights correspond to the revealed comparative advantage of each region in a given industry (or sector, technology, ...).

Usage

```
prody(mat, vec)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
vec	A vector that gives GDP, R&D, education or any other relevant regional attribute that will be used to compute the weighted average for each industry

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123

Hausmann, R., Hwang, J. & Rodrik, D. (2007) What you export matters, *Journal of economic growth* **12**: 1-25.

See Also

location.quotient

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## a vector of GDP of regions
vec <- c (5, 10, 15, 25, 50)
## run the function
prody (mat, vec)
```

RCA — *Compute an index of revealed comparative advantage (RCA) from regions - industries matrices*

Description

This function computes an index of revealed comparative advantage (RCA) from (incidence) regions - industries matrices. The numerator is the share of a given industry in a given region. The denominator is the share of a this industry in a larger economy (overall country for instance). This index is also refered to as a location quotient, or the Hoover-Balassa index.

Usage

```
RCA(mat, binary = FALSE)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns
`binary`	Logical; shall the returned output be a dichotomized version (0/1) of the RCA? Defaults to FALSE (the full values of the RCA will be returned), but can be set to TRUE (RCA above 1 will be set to 1 & RCA values below 1 will be set to 0)

Author(s)

Pierre-Alexandre Balland <`p.balland@uu.nl`>

References

Balassa, B. (1965) Trade Liberalization and Revealed Comparative Advantage, *The Manchester School* **33**: 99-123.

See Also

`location.quotient`

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
RCA (mat)
RCA (mat, binary = TRUE)
```

---

relatedness — *Compute the relatedness between entities (industries, technologies, ...) from their co-occurence matrix*

Description

This function computes the relatedness between entities (industries, technologies, ...) from their co-occurence (adjacency) matrix. Different normalization procedures are proposed following van Eck and Waltman (2009): association strength, cosine, Jaccard, and an adapted version of the association strength that we refer to as probability index.

Usage

```
relatedness(mat, method = "prob")
```

Arguments

`mat`	An adjacency matrix of co-occurences between entities (industries, technologies, cities...)
`method`	Which normalization method should be used to compute relatedness? Defaults to "prob", but it can be "association", "cosine" or "Jaccard"

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>
Joan Crespo <J.Crespo@uu.nl>
Mathieu Steijn <M.P.A.Steijn@uu.nl>

References

van Eck, N.J. and Waltman, L. (2009) How to normalize cooccurrence data? An analysis of some well-known similarity measures, *Journal of the American Society for Information Science and Technology* **60** (8): 1635-1651

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

Hidalgo, C.A., Klinger, B., Barabasi, A. and Hausmann, R. (2007) The product space conditions the development of nations, *Science* **317**: 482-487

Balland, P.A. (2016) Relatedness and the Geography of Innovation, in: R. Shearmur, C. Carrincazeaux and D. Doloreux (eds) Handbook on the Geographies of Innovation. Northampton, MA: Edward Elgar

Steijn, M.P.A. (2017) Improvement on the association strength: implementing probability measures based on combinations without repetition, *Working Paper, Utrecht University*

See Also

relatedness.density, co.occurence

Examples

```
## generate an industry - industry matrix in which cells give the number of co-occurences
## between two industries
set.seed(31)
mat <- matrix(sample(0:10,36,replace=T), ncol = 6)
mat[lower.tri(mat, diag = TRUE)] <- t(mat)[lower.tri(t(mat), diag = TRUE)]
rownames(mat) <- c ("I1", "I2", "I3", "I4", "I5", "I6")
colnames(mat) <- c ("I1", "I2", "I3", "I4", "I5", "I6")

## run the function
relatedness (mat)
relatedness (mat, method = "association")
relatedness (mat, method = "cosine")
relatedness (mat, method = "Jaccard")
```

---

relatedness.density *Compute the relatedness density between regions and industries from regions - industries matrices and industries - industries matrices*

Description

This function computes the relatedness density between regions and industries from regions - industries (incidence) matrices and industries - industries (adjacency) matrices

Usage

```
relatedness.density(mat, relatedness)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

relatedness, co.occurence

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1,16,replace=T), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
diag = TRUE)]
rownames(relatedness) <- c ("I1", "I2", "I3", "I4")
colnames(relatedness) <- c ("I1", "I2", "I3", "I4")

## run the function
relatedness.density (mat, relatedness)
```

---

relatedness.density.ext

Compute the relatedness density between regions and industries that are not part of the regional portfolio from regions - industries matrices and industries - industries matrices

---

Description

This function computes the relatedness density between regions and industries that are not part of the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices

Usage

```
relatedness.density.ext(mat, relatedness)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

relatedness, co.occurence

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1,16,replace=T), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
diag = TRUE)]
rownames(relatedness) <- c ("I1", "I2", "I3", "I4")
colnames(relatedness) <- c ("I1", "I2", "I3", "I4")

## run the function
relatedness.density.ext (mat, relatedness)
```

---

relatedness.density.ext.avg

Compute the average relatedness density of regions to industries that are not part of the regional portfolio from regions - industries matrices and industries - industries matrices

---

Description

This function computes the average relatedness density of regions to industries that are not part of the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices. This is the technological flexibility indicator proposed by Balland et al. (2015).

Usage

```
relatedness.density.ext.avg(mat, relatedness)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns
`relatedness`	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Balland P.A., Rigby, D., and Boschma, R. (2015) The Technological Resilience of U.S. Cities, *Cambridge Journal of Regions, Economy and Society*, **8** (2): 167-184

See Also

relatedness, relatedness.density, relatedness.density.ext, relatedness.density.int, avg.relatedness.density.int

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence
## of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1,16,replace=T), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
diag = TRUE)]
rownames(relatedness) <- c ("I1", "I2", "I3", "I4")
colnames(relatedness) <- c ("I1", "I2", "I3", "I4")

## run the function
relatedness.density.ext.avg (mat, relatedness)
```

---

relatedness.density.int

Compute the relatedness density between regions and industries that are part of the regional portfolio from regions - industries matrices and industries - industries matrices

---

Description

This function computes the relatedness density between regions and industries that are part of the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices

Usage

```
relatedness.density.int(mat, relatedness)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
relatedness	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Boschma, R., Heimeriks, G. and Balland, P.A. (2014) Scientific Knowledge Dynamics and Relatedness in Bio-Tech Cities, *Research Policy* **43** (1): 107-114

See Also

relatedness, co.occurence

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence
## of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1,16,replace=T), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
diag = TRUE)]
rownames(relatedness) <- c ("I1", "I2", "I3", "I4")
colnames(relatedness) <- c ("I1", "I2", "I3", "I4")

## run the function
relatedness.density.int (mat, relatedness)
```

`relatedness.density.int.avg`
Compute the average relatedness density within the regional portfolio from regions - industries matrices and industries - industries matrices

---

Description

This function computes the average relatedness density within the regional portfolio from regions - industries (incidence) matrices and industries - industries (adjacency) matrices. This is a measure of the technological coherence of the regional industrial structure.

Usage

```
relatedness.density.int.avg(mat, relatedness)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns
`relatedness`	An adjacency industry - industry matrix indicating the degree of relatedness between industries

Author(s)

Pierre-Alexandre Balland `<p.balland@uu.nl>`

References

Boschma, R., Balland, P.A. and Kogler, D. (2015) Relatedness and Technological Change in Cities: The rise and fall of technological knowledge in U.S. metropolitan areas from 1981 to 2010, *Industrial and Corporate Change* **24** (1): 223-250

Balland P.A., Rigby, D., and Boschma, R. (2015) The Technological Resilience of U.S. Cities, *Cambridge Journal of Regions, Economy and Society*, **8** (2): 167-184

See Also

`relatedness`, `relatedness.density`, `relatedness.density.ext`, `relatedness.density.int`, `avg.relatedness.density.ext`

Examples

```
## generate a region - industry matrix in which cells represent the presence/absence
## of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## generate an industry - industry matrix in which cells indicate if two industries are
## related (1) or not (0)
relatedness <- matrix(sample(0:1,16,replace=T), ncol = 4)
relatedness[lower.tri(relatedness, diag = TRUE)] <- t(relatedness)[lower.tri(t(relatedness),
```

```
diag = TRUE)]
rownames(relatedness) <- c ("I1", "I2", "I3", "I4")
colnames(relatedness) <- c ("I1", "I2", "I3", "I4")

## run the function
relatedness.density.int.avg (mat, relatedness)
```

---

spec.coeff *Compute the Hoover coefficient of specialization from regions - industries matrices*

Description

This function computes the Hoover coefficient of specialization from regions - industries matrices. The higher the coefficient, the greater the regional specialization. This index is closely related to the Krugman specialisation index.

Usage

```
spec.coeff(mat)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hoover, E.M. and Giarratani, F. (1985) *An Introduction to Regional Economics*. 3rd edition. New York: Alfred A. Knopf (see table 9-4 in particular)

See Also

Krugman.index

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
spec.coeff (mat)
```

TCI *Compute an index of knowledge complexity of industries using the eigenvector method*

Description

This function computes an index of knowledge complexity of industries using the eigenvector method from regions - industries (incidence) matrices. Technically, the function returns the eigenvector associated with the second largest eigenvalue of the projected industry - industry matrix.

Usage

```
TCI(mat, RCA = FALSE)
```

Arguments

mat	An incidence matrix with regions in rows and industries in columns
RCA	Logical; should the index of relative comparative advantage (RCA - also refered to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

References

Hidalgo, C. and Hausmann, R. (2009) The building blocks of economic complexity, *Proceedings of the National Academy of Sciences* **106**: 10570 - 10575.

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

location.quotient, ubiquity, diversity, MORc, KCI, MORt

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
TCI (mat, RCA = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
```

```
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
TCI (mat)

## generate the simple network of Hidalgo and Hausmann (2009) presented p.11 (Fig. S4)
countries <- c("C1", "C1", "C1", "C1", "C2", "C3", "C3", "C4")
products <- c("P1","P2", "P3", "P4", "P2", "P3", "P4", "P4")
data <- data.frame(countries, products)
data$freq <- 1
mat <- get.matrix (data)

## run the function
TCI (mat)
```

`ubiquity` *Compute a simple measure of ubiquity of industries*

Description

This function computes a simple measure of ubiquity of industries by counting the number of regions in which an industry can be found (location quotient > 1) from regions - industries (incidence) matrices

Usage

```
ubiquity(mat, RCA = FALSE)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns
`RCA`	Logical; should the index of relative comparative advantage (RCA - also refered to as location quotient) first be computed? Defaults to FALSE (a binary matrix - 0/1 - is expected as an input), but can be set to TRUE if the index of relative comparative advantage first needs to be computed

Author(s)

Pierre-Alexandre Balland <`p.balland@uu.nl`>

References

Balland, P.A. and Rigby, D. (2017) The Geography of Complex Knowledge, *Economic Geography* **93** (1): 1-23.

See Also

`diversity` `location.quotient`

Examples

```
## generate a region - industry matrix with full count
set.seed(31)
mat <- matrix(sample(0:10,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
ubiquity (mat, RCA = TRUE)

## generate a region - industry matrix in which cells represent the presence/absence of a RCA
set.seed(31)
mat <- matrix(sample(0:1,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## run the function
ubiquity (mat)
```

`weighted.avg` *Compute a weighted average of regions or industries from regions - industries matrices*

Description

This function computes a weighted average of regions or industries from (incidence) regions - industries matrices.

Usage

```
weighted.avg(mat, vec, reg = T)
```

Arguments

`mat`	An incidence matrix with regions in rows and industries in columns
`vec`	A vector that will be used to compute the weighted average for each industry/region
`reg`	Logical; Shall the weighted average for regions be returned? Default to TRUE (requires a vector of industry value) but can be set to FALSE (requires a vector of region value) if the weighted average for industries should be returned

Author(s)

Pierre-Alexandre Balland <p.balland@uu.nl>

See Also

location.quotient

Examples

```
## generate a region - industry matrix
set.seed(31)
mat <- matrix(sample(0:100,20,replace=T), ncol = 4)
rownames(mat) <- c ("R1", "R2", "R3", "R4", "R5")
colnames(mat) <- c ("I1", "I2", "I3", "I4")

## a vector for regions will be used to computed the weighted average of industries
vec <- c (5, 10, 15, 25, 50)
## run the function
weighted.avg (mat, vec, reg = F)

## a vector for industries will be used to computed the weighted average of regions
vec <- c (5, 10, 15, 25)
## run the function
weighted.avg (mat, vec, reg = T)
```

---

zScore — *Compute the z-score between technologies from an incidence matrix*

Description

This function computes the z-score between pairs of technologies from a patent-technology incidence matrix. The z-score is a measure to analyze the co-occurrence of technologies in patent documents (i.e. knowledge combination). It compares the observed number of co-occurrences to what would be expected under the hypothesis that combination is random. A positive z-score indicates a typical co-occurrence which has occurred multiple times before. In contrast, a negative z-socre indicates an atypical co-occurrence. The z-score has been used to estimate the degree of novelty of patents (Kim 2016), scientific publications (Uzzi et al. 2013) or the relatedness between industries (Teece et al. 1994).

Usage

```
zScore(mat)
```

Arguments

`mat`	A patent-technology incidence matrix with patents in rows and technologies in columns

Author(s)

Lars Mewes <mewes@wigeo.uni-hannover.de>

References

Kim, D., Cerigo, D. B., Jeong, H., and Youn, H. (2016). Technological novelty proile and invention's future impact. *EPJ Data Science*, **5(1)**:1–15

Teece, D. J., Rumelt, R., Dosi, G., and Winter, S. (1994). Understanding corporate coherence. Theory and evidence. *Journal of Economic Behavior and Organization*, **23(1)**:1–30

Uzzi, B., Mukherjee, S., Stringer, M., and Jones, B. (2013). Atypical Combinations and Scientific Impact. *Science*, **342(6157)**:468–472

See Also

relatedness.density, co.occurence

Examples

```
## Generate a toy incidence matrix
set.seed(2210)
techs <- paste0("T", seq(1, 5))
techs <- sample(techs, 50, replace = TRUE)
patents <- paste0("P", seq(1, 20))
patents <- sort(sample(patents, 50, replace = TRUE))
dat <- data.frame(patents, techs)
dat <- unique(dat)
mat <- as.matrix(table(dat$patents, dat$techs))

## run the function
zScore(mat)
```

Index