Package ‘ioanalysis’

August 7, 2015

Title Input-Output Analysis

Version 0.1

Author Ignacio Sarmiento-Barbieri

Maintainer Ignacio Sarmiento-Barbieri <srmntbr2@illinois.edu>

Description Input and Output Analysis.

Depends R (>= 3.1.1), ggplot2, xlsx, Matrix

License GNU GENERAL PUBLIC LICENSE

LazyData true

R topics documented:

agg.matrix 1
extraction 2
gosh.inv 3
influence 4
key.sector 5
leontief.inv 7
multipliers 8
upstream 10
vs.io 11
vs.ki 12

Index **14**

`agg.matrix` *Aggregation Matrix*

Description

Aggregates Input-Output Matrices. Construct an aggregation matrix of kxn dimension.

Usage

```
agg.matrix(mat,d.mip)
```

Arguments

`mat`	Matrix. How sectors should be aggregated. First column should be the new sector in order, the remaining columns the sectors that are going to be aggregated
`dimcol`	Numeric. Column dimension

Value

Returns matrix to be pre and post multiplied to aggregate sectors.

Author(s)

Ignacio Sarmiento-Barbieri

References

Nazara, Suahasil & Guo, Dong & Hewings, Geoffrey J.D., & Dridi, Chokri, 2003. *PyIO. Input-Output Analysis with Python*. REAL Discussion Paper 03-T-23. University of Illinois at Urbana-Champaign. (http://www.real.illinois.edu/d-paper/03/03-T-23.pdf)

Examples

```
temp<-matrix(sample(1:30), 10)
temp[,1]<-seq(1:10)

S<-agg.matrix(temp,40)
```

`extraction` *Implements Extraction Method*

Description

Applies the Extraction method as outlined in Dietzenbacher et al. (1993)

Usage

```
extraction(mip, X, f, v, forward=FALSE, write.xlsx=FALSE, name="Extraction.xlsx")
```

Arguments

`mip`	Matrix. Input output matrix
`X`	Vector. Input in each column
`f`	Vector. Final demand vector
`v`	Vector. Primary input vector
`write.xlsx`	Logical. If TRUE writes an excel file
`name`	String. Name of the excel file

Details

The matrix should be read cautiously. Columns indicate the sector under analysis. So the first column gives teh impact when sector 1 is hypthetically extracted from the system. Thus A[1,1] is the impact to sector 1 when that sector is hypothetically extracted. A[1,2] gives the impact to sector 1 when sector 2 was hypothetically extracted. A[2,1] represents the impact to sector 2 when sector 1 is hypothetically extracted and so on.

The diagonal of the output matrix represents Dietzenbacher's feedback effect.

Value

Returns a matrix with the Extraction matrix

References

Dietzenbacher Erik & van der Linden Jan A. & Steenge Alben E. (1993). *The Regional Extraction Method: EC Input–Output Comparisons*. Economic Systems Research. Vol. 5, Iss. 2, 1993

Nazara, Suahasil & Guo, Dong & Hewings, Geoffrey J.D., & Dridi, Chokri, 2003. *PyIO. Input-Output Analysis with Python*. REAL Discussion Paper 03-T-23. University of Illinois at Urbana-Champaign. (http://www.real.illinois.edu/d-paper/03/03-T-23.pdf)

Examples

```
#Follows the example in PyIO 2.0 Quick Start

mip<-matrix(c(16,5,24,0,6,17,10,0,7,17,11,48,26,0,8,0,43,82,33,13,17,81,51,4,35,9,93,7,19,99,30,2,19,20,19,
mip

X<-c(700,320,607,432,375,345,561,187)


f<-c(622,203,283,138,220,75,349,78)

extraction(mip,X,f, v,forward=FALSE,write.xlsx=FALSE, name="Extraction.xlsx")
```

`gosh.inv` *Gosh Inverse*

Description

Computes the Goshian Inverse

Usage

```
gosh.inv(mip, X, write.xlsx=FALSE, name="Gosh_Inv.xlsx")
```

Arguments

`mip`	Matrix. Input output matrix
`X`	Vector. Input in each column
`write.xlsx`	Logical. If TRUE writes an excel file
`name`	String. Name of the excel file

Details

The Goshian inverse is derived from the output allocation matrix table A=[a_ij] where

$$a_i j = z_{ij}/X_i$$

where z_ij is the input from i required in the production of j. X_i is the corresponding output in each row. The Goshian inverse is then computed as

$$(I - A)^{-1}$$

Value

Returns a data.frame with the Goshian Inverse

Author(s)

Ignacio Sarmiento-Barbieri

References

Nazara, Suahasil & Guo, Dong & Hewings, Geoffrey J.D., & Dridi, Chokri, 2003. *PyIO. Input-Output Analysis with Python*. REAL Discussion Paper 03-T-23. University of Illinois at Urbana-Champaign. (http://www.real.illinois.edu/d-paper/03/03-T-23.pdf)

Examples

```
mip<-mat_40x40[1:40,2:41] #Input-output coeffcients

X<-mat_40x40$DT.a.PB[1:40]  #Total output vector

G<-gosh.inv(mip,X,linkages=TRUE,write.xlsx = FALSE)
```

`influence` *Field of Influence*

Description

Computes the first order field of Influence

Usage

```
influence(mip,X, col,row, write.xlsx=FALSE, name="Field_of_Influence.xlsx")
```

Arguments

`mip`	Matrix. Input output matrix
`X`	Vector. Input in each column
`col`	Numeric. Column where the change is occurring
`row`	Numeric. Row where the change is occurring
`write.xlsx`	Logical. If TRUE writes an excel file
`name`	String. Name of the excel file

Details

The Field of influence assesses the changes in the Leontief inverse matrix resulting from the change in one or more direct input coefficients in the inverse Leontief matrix.

The first order field of influence deals with a single change in one elements of the coefficient matrix.

Value

Returns a matrix.

Author(s)

Ignacio Sarmiento-Barbieri

References

Nazara, Suahasil & Guo, Dong & Hewings, Geoffrey J.D., & Dridi, Chokri, 2003. *PyIO. Input-Output Analysis with Python*. REAL Discussion Paper 03-T-23. University of Illinois at Urbana-Champaign. (http://www.real.illinois.edu/d-paper/03/03-T-23.pdf)

See Also

See Also leontief.inv

Examples

```
#Follows the example in PyIO 2.0 Quick Start

mip<-matrix(c(16,5,24,0,6,17,10,0,7,17,11,48,26,0,
              8,0,43,82,33,13,17,81,51,4,35,9,93,7,
              19,99,30,2,19,20,19,6,59,16,16,0,15,15,
              99,45,66,11,12,7,25,22,47,4,42,26,45,1,
              0,0,75,0,12,7,12,3), ncol=8, byrow=TRUE)

X<-c(700,320,607,432,375,345,561,187)

F<-influence(mip,X,1,1,write.xlsx = FALSE)
```

---

key.sector — *Impact Analysis: Backward and Forward linkages*

Description

Computation of Backward and Forward linkages. It aims to identify thos sectors whose economic activity exerts a greater than average influence on the whole economy. Key sectors are identified by calculating backard and forward linkages. Let

$$B = (I - A)^{-1} = [b_{ij}]$$

be the Leontief inverse matrix and let B_j and B_i be the column and row multipliers of this Leontief inverse. The sector j's backward linkage (BL_j) and forward linkage (FL_i) are defined as:

$$BL_j = \frac{\frac{1}{n}\sum_{i=1}^{n} b_{ij}}{\frac{1}{n^2}\sum_{j=1}^{n}\sum_{i=1}^{n} b_{ij}}$$

$$FL_i = \frac{\frac{1}{n}\sum_{j=1}^{n} b_{ij}}{\frac{1}{n^2}\sum_{j=1}^{n}\sum_{i=1}^{n} b_{ij}}$$

The usual interpretation is to propose that, if

$$BL_j > 1$$

, a unit change in final demand in sector j will generate an above-average increase in activity in the economy. Similarly, for

$$FL_i > 1$$

, it is asserted that a unit change in all sector's final demand would create an above average increase in sector i. Thus, a key sector is identified as one having both indices grater than one.

Usage

```
key.sector(mip, X, epsilon=0.1, key=TRUE, cutoff=1, write.xlsx=TRUE, name="Key_sector.xlsx")
```

Arguments

`mip`	Input-output matrix
`X`	Total input or output
`epsilon`	Replaces zeros in X
`key`	Logical. If TRUE identifies key sectors
`cutoff`	Cutoff above which are the key sectors
`write.xlsx`	Logical. If TRUE writes an excel file
`name`	String. Name of the excel file

Details

The function takes the sector names from the column names of the Input-output matrix. If `key=TRUE` it orders the sectors first by the Backward Linkage and second by Forward Linkage

Value

Returns a vector with the calculated Backward and Forward linkages for each sector

Author(s)

Ignacio Sarmiento-Barbieri

References

Nazara, Suahasil & Guo, Dong & Hewings, Geoffrey J.D., & Dridi, Chokri, 2003. "PyIO. Input-Output Analysis with Python". REAL Discussion Paper 03-T-23. University of Illinois at Urbana-Champaign. (http://www.real.illinois.edu/d-paper/03/03-T-23.pdf)

See Also

See Also `leontief.inv`

Examples

```
#Uses the 40x40 matrix included in the package

mip<-mat_40x40[1:40,2:41] #Input-output coeffcients

X<-mat_40x40$DT.a.PB[1:40]  #Total output vector

key<-key.sector(mip,X, key=FALSE, write.xlsx=FALSE)
```

`leontief.inv` *Leontief Inverse*

Description

Computes the Leontief Inverse and Backward and Forward Linkages

Usage

```
leontief.inv(mip, X, linkages=TRUE, write.xlsx=TRUE, name="Leontief_Inv.xlsx")
```

Arguments

`mip`	Matrix. Input output matrix
`X`	Vector. Input in each column
`linkages`	Logical. If TRUE computes Backward and Forward Linkages
`write.xlsx`	Logical. If TRUE writes an excel file
`name`	String. Name of the excel file

Details

The Leontief inverse is derived from the input-output table A=[a_ij] where

$$a_i j = z_i j / X_j$$

where z_ij is the input from i required in the production of j. X_j is the corresponding input in each coulumn. The Leontief inverse is then computed as

$$(I - A)^{-1}$$

The user has the option to compute the Backward and Forward Linkages for Key-sector Analysis. The sector j's Backward Linkage (BL_j) is defined as:

$$BL_j = \frac{\frac{1}{n}\sum_{i=1}^{n} b_{ij}}{\frac{1}{n^2}\sum_{j=1}^{n}\sum_{i=1}^{n} b_{ij}}$$

The sector j's Forward Linkage (FL_i) is defined as:

$$FL_i = \frac{\frac{1}{n}\sum_{j=1}^{n} b_{ij}}{\frac{1}{n^2}\sum_{j=1}^{n}\sum_{i=1}^{n} b_{ij}}$$

Value

Returns a data.frame with the Leontief Inverse

Author(s)

Ignacio Sarmiento-Barbieri

References

Nazara, Suahasil & Guo, Dong & Hewings, Geoffrey J.D., & Dridi, Chokri, 2003. *PyIO. Input-Output Analysis with Python*. REAL Discussion Paper 03-T-23. University of Illinois at Urbana-Champaign. (http://www.real.illinois.edu/d-paper/03/03-T-23.pdf)

Examples

```
mip<-mat_40x40[1:40,2:41] #Input-output coeffcients

X<-mat_40x40$DT.a.PB[1:40]  #Total output vector

L<-leontief.inv(mip,X,linkages=TRUE,write.xlsx = FALSE)
```

`multipliers` *Multiplier Analysis*

Description

According to Nazara et al. (2003) and Blair and Miller (2009) four multipliers can be calculated from an input-output matrix: output multiplier, input multiplier, income multiplier and employment multiplier.

Output multiplier: it is computed from the Leontief inverse. Let

$$B = [b_{ij}]$$

be the Leontief inverse matrix the output multiplier for sector j,

$$O_j = \sum_{i=1}^{n} b_{ij}$$

Input multiplier: it is computed from the Goshian inverse. Let

$$G = [g_{ij}]$$

be the Goshian inverse matrix the input multiplier for sector j,

$$I_j = \sum_{i=1}^{n} g_{ij}$$

Income multiplier: the calculation of this multiplier requires a wage vector (z) to calculate the household input coefficient (a):

$$a_{n+1,i} = \frac{z_{n+1,i}}{X_i}$$

with the Leontief inverse, the household income multiplier for sector j is

$$H_j = \sum_{i=1}^{n} a_{n+1,i} b_{ij}$$

Employment multiplier: the calculation of this multiplier requires a sectoral employment vector (e) to calculate the labor input coefficient (w):

$$w_{n+1,i} = \frac{e_i}{X_i}$$

with the Leontief inverse, the employment multiplier for sector j is

$$E_j = \sum_{i=1}^{n} w_{n+1,i} b_{ij}$$

Usage

```
multipliers(mip, X, z, e,write.xlsx=TRUE, name="output_multiplier.xlsx")
```

Arguments

`mip`	Input-output matrix
`X`	Vector. Total input or output
`z`	Vector. Household input coefficient
`e`	Vector. If TRUE identifies key sectors
`write.xlsx`	Logical. If TRUE writes an excel file
`name`	String. Name of the excel file

Value

Returns a data frame with the calculated multipliers for each sector

Author(s)

Ignacio Sarmiento-Barbieri

References

Blair, P.D. and Miller, R.E. (2009). "Input-Output Analysis: Foundations and Extensions". Cambridge University Press

Nazara, Suahasil & Guo, Dong & Hewings, Geoffrey J.D., & Dridi, Chokri, 2003. "PyIO. Input-Output Analysis with Python". REAL Discussion Paper 03-T-23. University of Illinois at Urbana-Champaign. (`http://www.real.illinois.edu/d-paper/03/03-T-23.pdf`)

Wu, P.C (2009). "PyIO 2.0 Quick Start". (`http://www.real.illinois.edu/pyio/`)

See Also

See Also `leontief.inv` `gosh.inv`

Examples

```
#Follows the example in PyIO 2.0 Quick Start


mip<-matrix(c(16,5,24,0,6,17,10,0,7,17,11,48,26,0,
              8,0,43,82,33,13,17,81,51,4,35,9,93,7,
              19,99,30,2,19,20,19,6,59,16,16,0,15,15,
              99,45,66,11,12,7,25,22,47,4,42,26,45,1,
              0,0,75,0,12,7,12,3), ncol=8, byrow=TRUE)

X<-c(700,320,607,432,375,345,561,187)

e<-c(10,20,30,52,10,75,51,40)

z<-c(29870.9,18720,66563.8,2607,19007.7,69883,10194.2,173.1)

L<-multipliers(mip=mip,X=X,z=z,e=e)
```

`upstream` *upstreamness*

Description

Measures upstreamness as of equation 9 in Antras et al. (2012)

Usage

```
upstream(linv,y,x,m, write.xlsx=TRUE, name="Upstream.xlsx")
```

Arguments

`linv`	Matrix. Input output matrix
`y`	Output Vector
`x`	Exports Vector
`m`	Imports Vector
`write.xlsx`	Logical. if TRUE writes an excel file
`name`	String. name of the excel file

Author(s)

Ignacio Sarmiento-Barbieri

References

Pol Antràs & Davin Chor & Thibault Fally & Russell Hillberry, 2012. *Measuring the Upstreamness of Production and Trade Flows*. NBER Working Papers 17819, National Bureau of Economic Research, Inc.

vs.io *VS share of total exports*

Description

Vertical Specialization (VS) share of total exports. The formula for VS as a share fo total exports for country k is

$$VS\ share\ of\ total\ exports = \frac{VS}{X_k} = \frac{A*L*X}{X_k}$$

where A is the nxn imported coefficient matrix,L is the Leontief inverse, X is an nx1 vector of exports, n is the number of secotrs, adn Xk is the sum of exports across the n sectors.

Usage

```
vs.io(imp,exp,leon=1,namesector, write.xlsx=TRUE, name="Level_Verticalization.xlsx")
```

Arguments

`imp`	Is the nxn imported coefficient matrix
`exp`	Numeric Vector. Exports
`leon`	Leontief Inverse as output of funciton leontief.inv()
`namesector`	Character. Name of Sector i
`write.xlsx`	Logical. if TRUE writes an excel file
`name`	String. name of the excel file

Value

Returns a data frame with the A*L*X product, the total exports and the level of verticalization for each sector. Total.exp. returns a scalar, equals to the sum of exports across sectors

Author(s)

Ignacio Sarmiento-Barbieri

References

Hummels, David & Ishii, Jun & Yi, Kei-Mu, 2001. *The nature and growth of vertical specialization in world trade*. Journal of International Economics, Elsevier, vol. 54(1), pages 75-96, June.

See Also

See Also `leontief.inv`

Examples

```
#Uses the 40x40 matrix included in the package

exp<-mat_40x40$X[1:40] #Exports Vector
class(exp) #the class is numeric
imp<-mat_imp_40x40[1:40,2:41]/mat_40x40$DT.a.PB[1:40] #Imports Coef Matrix
class(imp) #the class is data.frame
namesector<-colnames(mat_40x40[2:41])
class(namesector) #character

hum<-vs.io(imp,exp,namesector, write.xlsx=FALSE)

#Using the Leontief Inverse
leon<-leontief.inv(mat_40x40[1:40,2:41], mat_40x40$DT.a.PB[1:40])[,1:40]
hum2<-vs.io(imp,exp,namesector,leon=leon, write.xlsx=FALSE)
```

---

vs.ki *Vertical Specialization*

Description

VS is the imported input content of exports for country k in sector i. The Vertical specialization chain is VS_ki=(imported intermediats/gross output)*exports. The first term is the share of imported inputs into gross production. Multiplying this ratio by the amount that iseported provides the monetary value for the imported input content of exports.

Usage

```
vs.ki(imp,exp,out,namesector, write.xlsx=TRUE, name="Level_Verticalization.xlsx")
```

Arguments

`imp`	Vector. Imported intermediates of sector i
`exp`	Vector. Exports
`out`	Vector. Gross Output
`namesector`	String. Name of Sector i
`write.xlsx`	Logical. if TRUE writes an excel file
`name`	String. name of the excel file

Details

The dimension of the four arguments must coincide.

Value

Returns a data frame

Author(s)

Ignacio Sarmiento-Barbieri

References

Hummels, David & Ishii, Jun & Yi, Kei-Mu, 2001. *The nature and growth of vertical specialization in world trade*. Journal of International Economics, Elsevier, vol. 54(1), pages 75-96, June.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

Index

agg.matrix, 1

extraction, 2

gosh.inv, 3, *9*

influence, 4
influnce *(*influence*)*, 4

key.sector, 5

leontief.inv, *5*, *7*, 7, *9*, *11*

multipliers, 8

upstream, 10

vs.io, 11
vs.ki, 12