Libreria utilMIO

Francisco Parra

25 de marzo de 2018

Introducción

La libreria contiene diversas funciones para regionalizar, modelizar y actualizar marcos imput-output.

Para la regionalización de una TIO a partir de otra de ambito superior emplean utilizando una mínima información se emplean los cocientes de localización (LQ) . Entre ellos destaca el método FLQ (Flegg Location Quantion) o a su versión modificada AFLQ.

Para evaluar la similaridad entre tablas, hay varios metodos: WAPE, SAPE,

Una vez obtenidas dichas estimaciones, las mismas son susceptibles de posteriores ajustes con arreglo a la información disponible (real o incluso pronosticada). Las técnicas tradicionales, como el RAS básico, no son efectivas ya que para su aplicación, ya que necesitan conocer los márgenes de la matriz de consumos intermedios, de ahí que será preciso acudir a técnicas más complejas. En este sentido, se puede acudir a procedimientos de carácter global, tal como procede Eurostat. Este instituto emplea una técnica, conocida como el eurométodo (EU), diseñada por Beutel (2002), para elaborar el marco input-output (tabla simétrica y tablas supply-use) homogéneamente para los distintos estados que forman parte de la Unión Europea (Eurostat, 2008).

Finalmente, se incluye una modelización de impactos que utiliza los multiplicadores I y II del modelo de Leontief.

Regionalización por coeficientes de localización.

La regionalización de una TIO a partir del LQ surge de la hipótesis adoptada por Jensen et al. (1979), en la que se admite que los coeficientes técnicos regionales (a_{ij}^R) derivan de los nacionales (a_{ij}^N) a partir de un efecto multiplicativo surgido de un factor de participación dentro del comercio regional LQ_{ij}:

$$a_{ij}^R = a_{ij}^N LQ_{ij}, i, j = 1,2, \dots n.$$

Los subíndices i y j hacen referencia a los sectores suministradores y compradores,respectivamente.

Según Jensen et al. (1979), a los coeficientes técnicos regionales se les impone una restricción primordial dada por el siguiente criterio:

$a_{ij}^{R} = a_{ij}^{N} LQ_{ij}$, si $LQ_{ij} \leq 1$

$a_{ij}^{R} = a_{ij}^{N}$, si $LQ_{ij} > 1$

Lo cual significa que si la región es autosuficiente, el LQ es mayor que la unidad, entonces el coeficiente regional es exactamente el coeficiente asociado a la matriz de consumos intermedios nacional. En cambio, si la región es importadora neta, lo que indica que el LQ sea menor que la unidad, el coeficiente regional será inferior al nacional.

Los Los coeficientes de localización simple (SLQ) verifican la aportación de la industria de una región con la contribución de la misma industria al total (de la nación). Su expresión genérica es la siguiente:

$$SLQ_i = \frac{\frac{x_i^R}{x^R}}{\frac{x_i^N}{x^N}}$$

en donde, x_i^R es la producción del sector i en la región R, x^R es la producción (total) de la región R, x_i^N es la producción del sector i en el total del país y, por último, x^N es la producción (total) del país.

En la práctica la obtención de estos coeficientes es muy sencilla.Según Flegg et al. (1997) y Fuentes (2002), dichos coeficientes son algo imprecisos dado que, generalmente, los resultados sobreestiman la producción regional de algunas industrias. Por este motivo, otras fórmulas como la FLQ o la AFLQ corrigen estos problemas y, en consecuencia, optimizan el proceso de generación de TIOs regionales.

Función CILQ(a,b)

Los coeficientes de localización interindustrial (CILQ) cuantifican, para una determinada región, la importancia relativa de la industria suministradora i respecto a la industria compradora j, para mayor detalle véase Schaffer y Chu (1969). Su término genérico se escribe del siguiente modo:

$$CILQ_{ij} = \frac{SLQ - i}{SLQ_j}$$

La función CILQ opera sobre dos vectores de la misma dimension en los que están los valores añadidos de la región (a) y la nación (b). Devuelve una matriz n*n

```
#Leemos los valores añadidos brutos de Cantabria y España
a=c(170,2227,403,821,4896,2484) # VAB Cantabria 2012
```

```r
b=c(24019,129248,36320,63521,484087,216831) # VAB España 2012


CILQ=function(a,b) {
n=length(a)
# Calculamos SLQ
SLQ=(a/sum(a))/(b/sum(b))
# Calculamos CILQ
CILQ=t(matrix(SLQ,ncol=n,nrow=n)%*%diag(1/SLQ))
as.matrix(CILQ)}

CILQ(a,b)

##           [,1]     [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 1.0000000 2.434459 1.5677081 1.8261306 1.4289729 1.6185876
## [2,] 0.4107690 1.000000 0.6439658 0.7501178 0.5869777 0.6648656
## [3,] 0.6378738 1.552877 1.0000000 1.1648409 0.9115044 1.0324547
## [4,] 0.5476060 1.333124 0.8584863 1.0000000 0.7825141 0.8863482
## [5,] 0.6998033 1.703642 1.0970874 1.2779323 1.0000000 1.1326930
## [6,] 0.6178226 1.504063 0.9685655 1.1282247 0.8828517 1.0000000
```

Función MOSLQ(a,b)

El $CILQ_{ij}$ basa en una hipótesis engañosa que está implícita, si $i = j$ implica que todos los sectores pueden satisfacer toda la demanda de su propio sector a nivel local, indistintamente del tamaño del sector. Morrison y Smith (1974) modificaron el CILQ correspondiente a los elementos de la diagonal principal de la siguiente forma:

$MOSLQ_{ij} = CILQ_{ij}SLQ_i$ si (i=j)

La función MOSLQ opera sobre dos vectores de la misma dimension en los que están los valores añadidos de la región (a) y la nación (b). Devuelve una matriz n*n

```r
MOSLQ=function (a,b) {
n=length(a)
# Calculamos SLQ
SLQ=(a/sum(a))/(b/sum(b))
# Calculamos CILQ
CILQ1=CILQ(a,b)
MOSLQ=CILQ1
for(i in seq(1:n)) {MOSLQ[i,i]=as.numeric(SLQ[i])}
as.matrix(MOSLQ)}

MOSLQ(a,b)

##           [,1]     [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.6137932 2.434459 1.5677081 1.8261306 1.4289729 1.6185876
## [2,] 0.4107690 1.494254 0.6439658 0.7501178 0.5869777 0.6648656
## [3,] 0.6378738 1.552877 0.9622485 1.1648409 0.9115044 1.0324547
## [4,] 0.5476060 1.333124 0.8584863 1.1208665 0.7825141 0.8863482
```

```
## [5,] 0.6998033 1.703642 1.0970874 1.2779323 0.8770938 1.1326930
## [6,] 0.6178226 1.504063 0.9685655 1.1282247 0.8828517 0.9934780
```

Función RLQ (a,b)

Otra propuesta es la sugerida por Round (1978), simbolizada normalmente mediante la abreviatura RLQ. Su expresión es del siguiente modo:

$$RLQ_{ij} = \frac{SLQ_i}{log_2(1 + SLQ_j)}$$

La función RLQ opera sobre dos vectores de la misma dimension en los que están los valores añadidos de la región (a) y la nación (b). Devuelve una matriz n*n

```
# Calculamos RLQ

RLQ=function(a,b) {
n=length(a)
# Calculamos SLQ
SLQ=(a/sum(a))/(b/sum(b))
RLQ=t(matrix(as.numeric(SLQ),ncol=n,nrow=n)%*%diag(1/log2(1+SLQ)))
as.matrix(RLQ)}

RLQ(a,b)

##           [,1]     [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.8889682 2.164156 1.3936427 1.6233721 1.2703115 1.4388730
## [2,] 0.4654856 1.133205 0.7297455 0.8500374 0.6651662 0.7534292
## [3,] 0.6311447 1.536496 0.9894507 1.1525527 0.9018887 1.0215631
## [4,] 0.5658886 1.377632 0.8871481 1.0333864 0.8086394 0.9159402
## [5,] 0.6756111 1.644747 1.0591610 1.2337541 0.9654299 1.0935357
## [6,] 0.6166992 1.501329 0.9668044 1.1261733 0.8812465 0.9981818
```

Función FLQ (a,b,c)

Flegg y Webber (1995), para superar los problemas que caracterizaban a los procedimientos anteriores, especialmente la sobrestimación de la autosuficiencia de los distintos sectores productivos, propusieron el coeficiente FLQ_{ij}, cuyos principales elementos de este procedimiento son, por un lado, los $CILQ_{ij}$ y, por otro, el rol atribuido al tamaño de la economía regional:

$$FLQ_{ij} = CILQ_{ij}[log_2(1 - \frac{x^R}{x^N})]^\delta$$

La función FLQ opera sobre dos vectores de la misma dimension en los que están los valores añadidos de la región (a) y la nación (b), y el coeficiente delta (c). Devuelve una matriz n*n

```
# Calculamos los FLQ
```

```
FLQ= function(a,b,c) {
coef=log2(1+(a/b))^c
CILQ1=CILQ(a,b)
FLQ=CILQ1%*%diag(as.numeric(coef))
as.matrix(FLQ)}

FLQ(a,b,0.10)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.6320532 1.6810398 1.0362362 1.2254983 0.9358679 1.0732703
## [2,] 0.2596278 0.6905189 0.4256536 0.5033967 0.3844255 0.4408661
## [3,] 0.4031702 1.0722913 0.6609879 0.7817133 0.5969656 0.6846111
## [4,] 0.3461161 0.9205474 0.5674491 0.6710902 0.5124868 0.5877292
## [5,] 0.4423129 1.1763972 0.7251615 0.8576078 0.6549235 0.7510781
## [6,] 0.3904967 1.0385843 0.6402101 0.7571405 0.5782003 0.6630906
```

Funcion RegioFLQ(a,b,c,N)

Obtiene los coeficientes técnicos regionales a partir de los coeficientes nacionales (A.N), utilizando los coeficientes FLQ.

Los coeficientes técnicos regionales se calcularían entonces:

$a_{ij}^{R} = a_{ij}^{N} FLQ_{ij}$, si $CILQ_{ij} \leq 1$

$a_{ij}^{R} = a_{ij}^{N}$, si $CILQ_{ij} > 1$

La función RegioFLQ opera sobre dos vectores de la misma dimension en los que están los valores añadidos de la región (a) y la nación (b), el coeficiente delta (c) y la matriz $n * n$ de los coeficientes tecnicos nacionales. Devuelve una matriz n*n

```
# lemos los coeficientes tecnicos nacionales a^N_{ij}
A.N=read.csv("AN.csv", header = FALSE, sep = ";", dec = ",")

RegioFLQ=function(a,b,c,N) {
# Bucle para obtener los coeficientes
n=length(a)
CILQ.0=CILQ(a,b)
CILQ.1=CILQ.0
FLQ.1=FLQ(a,b,c)
for (i in 1:n) {
  for (j in 1:n) { CILQ.0[i,j]=CILQ.1[i,j]>1
  }
}
for (i in 1:n) {
  for (j in 1:n) { CILQ.1[i,j]=CILQ.1[i,j]<=1
  }
}
A.R=as.matrix(N*CILQ.0+N*FLQ.1*CILQ.1,nrow=n)
as.matrix(A.R)}
```

```
RegioFLQ(a,b,0.10,A.N)
```

```
##                V1          V2          V3           V4          V5
## [1,] 0.0374523966 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.0898860796 0.370415628 0.115557918 0.1042609295 0.034409091
## [3,] 0.0073404486 0.050353148 0.180889866 0.0283924550 0.011430355
## [4,] 0.0009686747 0.004780369 0.006324446 0.1285162632 0.007319633
## [5,] 0.0203336291 0.106248881 0.096343601 0.1352887030 0.176590774
## [6,] 0.0012514444 0.007506077 0.005356770 0.0077104780 0.007845817
##               V6
## [1,] 0.001593652
## [2,] 0.037118612
## [3,] 0.016721470
## [4,] 0.005829162
## [5,] 0.126354291
## [6,] 0.030557197
```

Función AFLQ (a,b,c)

Posteriormente, Flegg y Webber (2000) responden con una reformulación, incorporando una corrección en relación a la especialización de las ramas compradoras:

$AFLQ_{ij} = FLQ_{ij} log_2(1 - SLQ_j)$ si $SLQ_j > 1$

$AFLQ_{ij} = FLQ_{ij}$ si $SLQ_j \leq 1$

```
# Bucle para obtener los AFLQ
AFLQ = function(a,b,c) {
FLQ.0=FLQ(a,b,c)
FLQ.1=FLQ.0
n=length(a)
# Calculamos SLQ
SLQ=(a/sum(a))/(b/sum(b))
for (i in 1:n) {
  for (j in 1:n) { FLQ.0[i,j]=SLQ[j]>1
  }
}
for (i in 1:n) {
  for (j in 1:n) { FLQ.1[i,j]=SLQ[j]<=1
  }
}
FLQ.1*FLQ(a,b,c)
AFLQ=log2(1+SLQ)
FLQ.0=FLQ.0%*%diag(as.numeric(AFLQ))
FLQ.0
AFLQ=FLQ.1*FLQ(a,b,c)+FLQ.0*AFLQ
as.matrix(AFLQ)}
AFLQ(a,b,0.10)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.6320532 0.9104407 1.0362362 0.7489054 0.9358679 1.0732703
## [2,] 0.2596278 1.7387281 0.4256536 1.4302336 0.3844255 0.4408661
## [3,] 0.4031702 1.2823569 0.6609879 1.0548342 0.5969656 0.6846111
## [4,] 0.3461161 1.4302336 0.5674491 1.1764738 0.5124868 0.5877292
## [5,] 0.4423129 1.1979567 0.7251615 0.9854088 0.6549235 0.7510781
## [6,] 0.3904967 1.3123947 0.6402101 1.0795426 0.5782003 0.6630906
```

Funcion RegioAFLQ(a,b,c,N)

Obtiene los coeficientes técnicos regionales a partir de los coeficientes nacionales (A.N), utilizando los coeficientes FLQ.

La función RegioAFLQ opera sobre dos vectores de la misma dimension en los que están los valores añadidos de la región (a) y la nación (b), el coeficiente delta (c) y la matriz $n * n$ de los coeficientes tecnicos nacionales. Devuelve una matriz n*n

```
RegioAFLQ=function(a,b,c,N) {
# Bucle para obtener los coeficientes
n=length(a)
CILQ.0=CILQ(a,b)
CILQ.1=CILQ.0
AFLQ.1=AFLQ(a,b,c)
for (i in 1:n) {
  for (j in 1:n) { CILQ.0[i,j]=CILQ.1[i,j]>1
  }
}
for (i in 1:n) {
  for (j in 1:n) { CILQ.1[i,j]=CILQ.1[i,j]<=1
  }
}
A.R=as.matrix(N*CILQ.0+N*AFLQ.1*CILQ.1,nrow=n)
as.matrix(A.R)}

RegioAFLQ(a,b,0.10,A.N)

##                V1          V2          V3           V4          V5
## [1,] 0.0374523966 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.0898860796 0.932707305 0.115557918 0.2962226379 0.034409091
## [3,] 0.0073404486 0.050353148 0.180889866 0.0283924550 0.011430355
## [4,] 0.0009686747 0.004780369 0.006324446 0.2252991163 0.007319633
## [5,] 0.0203336291 0.106248881 0.096343601 0.1352887030 0.176590774
## [6,] 0.0012514444 0.007506077 0.005356770 0.0077104780 0.007845817
##               V6
## [1,] 0.001593652
## [2,] 0.037118612
## [3,] 0.016721470
## [4,] 0.005829162
## [5,] 0.126354291
## [6,] 0.030557197
```

Funcion RegioMIO(a,b,N,c)

Regionaliza utilizando los siguientes metodos:

MOSLQ

RLQ

AFLQ

FLQ (por defecto)

La función RegioMIO opera sobre dos vectores de la misma dimension en los que están los valores añadidos de la región (a) y la nación (b), el coeficiente delta hay que especificarlo, por defecto utiliza 0.10, y la matriz $n * n$ de los coeficientes tecnicos nacionales. Devuelve una matriz n*n

```
RegioMIO=function(a,b,N,method="FLQ",delta=0.1) {
# Bucle para metodos
n=length(a)
D <- as.numeric(delta)
if(method=="MOSLQ") {
MOSLQ.0=MOSLQ(a,b)
MOSLQ.1=MOSLQ.0
for (i in 1:n) {
  for (j in 1:n) { MOSLQ.0[i,j]=MOSLQ.1[i,j]>1
  }
}
for (i in 1:n) {
  for (j in 1:n) { MOSLQ.1[i,j]=MOSLQ.1[i,j]<=1
  }
}
A.R=as.matrix(N*MOSLQ.0+N*MOSLQ(a,b)*MOSLQ.1,nrow=n)
as.matrix(A.R)} else {if(method=="RLQ") {
RLQ.0=RLQ(a,b)
RLQ.1=RLQ.0
for (i in 1:n) {
  for (j in 1:n) { RLQ.0[i,j]=RLQ.1[i,j]>1
  }
}
for (i in 1:n) {
  for (j in 1:n) { RLQ.1[i,j]=RLQ.1[i,j]<=1
  }
}
A.R=as.matrix(N*RLQ.0+N*RLQ(a,b)*RLQ.1,nrow=n)
as.matrix(A.R)} else {
  if(method=="AFLQ") {
  RegioAFLQ(a,b,D,N)} else {RegioFLQ(a,b,D,N)}
}
}
}
```

```r
RegioMIO(a,b,A.N)
```

```
##                V1          V2          V3           V4          V5
## [1,] 0.0374523966 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.0898860796 0.370415628 0.115557918 0.1042609295 0.034409091
## [3,] 0.0073404486 0.050353148 0.180889866 0.0283924550 0.011430355
## [4,] 0.0009686747 0.004780369 0.006324446 0.1285162632 0.007319633
## [5,] 0.0203336291 0.106248881 0.096343601 0.1352887030 0.176590774
## [6,] 0.0012514444 0.007506077 0.005356770 0.0077104780 0.007845817
##               V6
## [1,] 0.001593652
## [2,] 0.037118612
## [3,] 0.016721470
## [4,] 0.005829162
## [5,] 0.126354291
## [6,] 0.030557197
```

```r
RegioMIO(a,b,A.N,"MOSLQ")
```

```
##               V1          V2          V3           V4          V5
## [1,] 0.036370397 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.142212845 0.536430798 0.174826060 0.1553605375 0.052539102
## [3,] 0.011613657 0.050353148 0.263334615 0.0283924550 0.017452964
## [4,] 0.001532584 0.004780369 0.009568172 0.1915037190 0.011176318
## [5,] 0.032170757 0.106248881 0.096343601 0.1352887030 0.236495833
## [6,] 0.001979967 0.007506077 0.008104187 0.0077104780 0.011979747
##               V6
## [1,] 0.001593652
## [2,] 0.055978187
## [3,] 0.016721470
## [4,] 0.008790898
## [5,] 0.126354291
## [6,] 0.045782435
```

```r
RegioMIO(a,b,A.N,"RLQ")
```

```
##               V1          V2          V3           V4          V5
## [1,] 0.052675933 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.161156349 0.536430798 0.198113816 0.1760553835 0.059537588
## [3,] 0.011491141 0.050353148 0.270778931 0.0283924550 0.017268847
## [4,] 0.001583752 0.004780369 0.009887618 0.1915037190 0.011549455
## [5,] 0.031058611 0.106248881 0.096343601 0.1352887030 0.260314410
## [6,] 0.001976367 0.007506077 0.008089452 0.0077104780 0.011957965
##               V6
## [1,] 0.001593652
## [2,] 0.063434778
## [3,] 0.016721470
## [4,] 0.009084394
## [5,] 0.126354291
## [6,] 0.045999197
```

```r
RegioMIO(a,b,A.N,"FLQ")
```

```
##                V1          V2          V3           V4          V5
## [1,] 0.0374523966 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.0898860796 0.370415628 0.115557918 0.1042609295 0.034409091
## [3,] 0.0073404486 0.050353148 0.180889866 0.0283924550 0.011430355
## [4,] 0.0009686747 0.004780369 0.006324446 0.1285162632 0.007319633
## [5,] 0.0203336291 0.106248881 0.096343601 0.1352887030 0.176590774
## [6,] 0.0012514444 0.007506077 0.005356770 0.0077104780 0.007845817
##               V6
## [1,] 0.001593652
## [2,] 0.037118612
## [3,] 0.016721470
## [4,] 0.005829162
## [5,] 0.126354291
## [6,] 0.030557197
```

```r
RegioMIO(a,b,A.N,method="AFLQ",0.24)
```

```
##                V1          V2          V3           V4          V5
## [1,] 0.0197031051 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.0472876246 0.932707305 0.064724792 0.2962226379 0.019025670
## [3,] 0.0038616923 0.050353148 0.101317669 0.0283924550 0.006320137
## [4,] 0.0005096042 0.004780369 0.003542366 0.2252991163 0.004047213
## [5,] 0.0106971961 0.106248881 0.096343601 0.1352887030 0.097641577
## [6,] 0.0006583648 0.007506077 0.003000364 0.0077104780 0.004338154
##               V6
## [1,] 0.001593652
## [2,] 0.020883042
## [3,] 0.016721470
## [4,] 0.003279504
## [5,] 0.126354291
## [6,] 0.017191571
```

```r
RegioMIO(a,b,A.N,"AFLQ",delta="0.24")
```

```
##                V1          V2          V3           V4          V5
## [1,] 0.0197031051 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.0472876246 0.932707305 0.064724792 0.2962226379 0.019025670
## [3,] 0.0038616923 0.050353148 0.101317669 0.0283924550 0.006320137
## [4,] 0.0005096042 0.004780369 0.003542366 0.2252991163 0.004047213
## [5,] 0.0106971961 0.106248881 0.096343601 0.1352887030 0.097641577
## [6,] 0.0006583648 0.007506077 0.003000364 0.0077104780 0.004338154
##               V6
## [1,] 0.001593652
## [2,] 0.020883042
## [3,] 0.016721470
## [4,] 0.003279504
## [5,] 0.126354291
## [6,] 0.017191571
```

```
RegioMIO(a,b,A.N,method="RLQ")

##               V1          V2          V3           V4          V5
## [1,] 0.052675933 0.056145807 0.000390625 0.0000588844 0.003481128
## [2,] 0.161156349 0.536430798 0.198113816 0.1760553835 0.059537588
## [3,] 0.011491141 0.050353148 0.270778931 0.0283924550 0.017268847
## [4,] 0.001583752 0.004780369 0.009887618 0.1915037190 0.011549455
## [5,] 0.031058611 0.106248881 0.096343601 0.1352887030 0.260314410
## [6,] 0.001976367 0.007506077 0.008089452 0.0077104780 0.011957965
##               V6
## [1,] 0.001593652
## [2,] 0.063434778
## [3,] 0.016721470
## [4,] 0.009084394
## [5,] 0.126354291
## [6,] 0.045999197
```

Técnicas de Evaluación de Resultados

Cada método de actualización produce una proyección potencialmente diferente. Por tanto, es preciso evaluar su capacidad de predicción. Para ello es necesario disponer de estadísticos que nos permitan determinar si la proyección realizada se aproxima en mayor o menor medida al objetivo previsto.

Los estadísticos más habituales se basan en comparar la distancia entre los datos proyectados ($\hat{u}$) y los valores reales (u), utilizando diferentes medidas de distancia. Las medidas más básicas e intuitivas para medir la similitud de la bondad de ajuste de una matriz estimada frente a una teórica son (Valderas, 2015):

- media de los porcentajes de error en valor absoluto:

$$WAPE = \frac{\sum_i^m \sum_j^n \mid \hat{u}_{ij} - u_{ij} \mid}{\sum_i^m \sum_j^n u_{ij}} \times 100$$

Funcion wape(a,b)

calcula el wape de la tabla estimada a, frente a la b. Las tablas tienes que ser matrices o data.frames

```
# Lemos los coeficientes tecnicos regionales reales a^ {RR}_{ij}
A.RR=read.csv("AR.csv", header = FALSE, sep = ";", dec = ",")

wape=function(a,b) {
  sum(abs(a-b)/sum(b))
}

wape (FLQ(a,b,0.10),A.RR)

## [1] 14.58607
```

- media simétrica de los porcentajes de error en valor absoluto:

$$SWAPE = 200 \times \frac{\sum_i^m \sum_j^n u_{ij} \frac{|\hat{u}_{ij} - u_{ij}|}{|\hat{u}_{ij} + u_{ij}|}}{\sum_i^m \sum_j^n u_{ij}}$$

cuyo ρ − likelihood se obtiene como:

$$\rho - likelihood = 100 \times (1 - \frac{SWAPE}{200})$$

Funcion swape(a,b) y pswape(a,b)

calcula el swape de la tabla estimada a, frente a la b.Las tablas tienes que ser matrices o data.frames

```
swape=function(a,b) {
  200*sum(b*(abs(a-b)/abs(a+b))/sum(b))
}

pswape=function(a,b) {
  100*(1-(swape(a,b)/200))
}

swape (FLQ(a,b,0.10),A.RR)

## [1] 139.2571

pswape (FLQ(a,b,0.10),A.RR)

## [1] 30.37147
```

- media de los errores absolutos escalados:

$$MASE = m \times n \frac{\sum_i^m \sum_j^n |\hat{u}_{ij} - u_{ij}|}{\sum_i^m \sum_j^n |u_{ij} - \bar{u}|}$$

Funcion mase(a,b)

calcula el swape de la tabla estimada a, frente a la b.Las tablas tienes que ser matrices o data.frames

```
mase=function(a,b) {
  g=as.matrix(a)
  f=as.matrix(b)
  m=dim(g)[1]
  n=dim(g)[2]
 c=matrix(rep(mean(f),n*m),nrow=m)
  m*n*(sum(abs(g-f))/sum(abs(f-c)))
}
mase(FLQ(a,b,0.10),A.RR)
```

```
## [1] 539.5162

#library(MLmetrics)
#MASE(FLQ(a,b,0.10),A.RR)
```

- Psi de Kullback modificado:

$$\psi(U,\hat{U}) = \sum_i^m \sum_j^n u_{ij} \, ln \frac{| u_{ij} |}{\frac{| u_{ij} | + | \hat{u}_{ij} |}{2}} + \sum_i^m \sum_j^n \hat{u}_{ij} \, ln \frac{| \hat{u}_{ij} |}{\frac{| u_{ij} | + | \hat{u}_{ij} |}{2}}$$

cuyo ρ − likelihood se obtiene como:

$$\rho - psi = 100 \times \frac{\psi(U,\hat{U})}{ln2 \times \sum_i^m \sum_j^n | u_{ij} | + | \hat{u}_{ij} |}$$

```
Kpsi=function (a,b) {
  g=as.matrix(a)
  f=as.matrix(b)
  sum(g*log(abs(g)/((abs(g)+abs(f))/2)))+sum(f*log(abs(f)/((abs(g)+abs(f)
)/2)))
}
Kpsi(FLQ(a,b,0.10),A.RR)

## [1] 13.97414
```

- Indice de similaridad:

$$IS = 100 - 50(1 - r_{U,\hat{U}})$$

donde

$$r_{U,\hat{U}} = \frac{cov(U,\hat{U})}{S_U S_{\hat{U}}}$$

```
IS=function(a,b){
 g=c(as.matrix(a))
 f=c(as.matrix(b))
 100-50*(1-cor(g,f))
}
IS(FLQ(a,b,0.10),A.RR)

## [1] 41.81437
```

Buscar un Delta para el FLQ, utilizando las distancias: wape,swape, mase ó Kpsi

busca el mejor coeficiente delta para la función FLQ, con los siguitentes parámetros:

- Vr=valores añadidos regionales
- Vn=valores añadidos nacionales
- R=coeficientes interindustriales regionales
- N=coeficientes interindustriales nacionales
- method= wape,swape, mase ó Kpsi. Por defecto utiliza el wape

```
deltaFLQ=function(vr,vn,R,N,method="wape") {
coef=seq(0.01,1,by=0.01)
dist=NA
n=length(vr)
for (k in coef) {
FLQ.0=FLQ(vr,vn,k)
CILQ.0=CILQ(vr,vn)
CILQ.1=CILQ.0
for (i in 1:n) {
  for (j in 1:n) { CILQ.0[i,j]=CILQ.0[i,j]>1
  }
}
for (i in 1:n) {
  for (j in 1:n) { CILQ.1[i,j]=CILQ.0[i,j]<=1
  }
}
A.R=N*CILQ.0+N*FLQ.0*CILQ.1
m=ifelse(method=="wape",wape(A.R,R),ifelse(method=="swape",swape(A.R,R),i
felse(method=="mase",swape(A.R,R),ifelse(method=="Kpsi",Kpsi(A.R,R),wape(
A.R,R)))))
dist=c(dist,m)
}
RES=data.frame(coef,dist=dist[2:101])
delta=subset(RES,dist==min(RES$dist))
as.numeric(delta[1])}
deltaFLQ(a,b,A.RR,A.N)

## [1] 0.24

deltaFLQ(a,b,A.RR,A.N,method="swape")

## [1] 0.1

deltaFLQ(a,b,A.RR,A.N,method="mase")

## [1] 0.1

deltaFLQ(a,b,A.RR,A.N,method="Kpsi")

## [1] 0.22

deltaFLQ(a,b,A.RR,A.N,method="otro")

## [1] 0.24
```

Obtenemos el A.R correspondiente al valor más bajo de delta

```
deltaAFLQ=function(vr,vn,R,N,method="wape") {
coef=seq(0.01,1,by=0.01)
dist=NA
n=length(vr)
for (k in coef) {
FLQ.0=AFLQ(vr,vn,k)
CILQ.0=CILQ(vr,vn)
CILQ.1=CILQ.0
for (i in 1:n) {
  for (j in 1:n) { CILQ.0[i,j]=CILQ.0[i,j]>1
  }
}
for (i in 1:n) {
  for (j in 1:n) { CILQ.1[i,j]=CILQ.0[i,j]<=1
  }
}
A.R=N*CILQ.0+N*FLQ.0*CILQ.1
m=ifelse(method=="wape",wape(A.R,R),ifelse(method=="swape",swape(A.R,R),i
felse(method=="mase",swape(A.R,R),ifelse(method=="Kpsi",Kpsi(A.R,R),wape(
A.R,R)))))
dist=c(dist,m)
}
RES=data.frame(coef,dist=dist[2:101])
delta=subset(RES,dist==min(RES$dist))
as.numeric(delta[1])}
deltaAFLQ(a,b,A.RR,A.N)

## [1] 0.11

deltaAFLQ(a,b,A.RR,A.N,method="swape")

## [1] 0.1

deltaAFLQ(a,b,A.RR,A.N,method="mase")

## [1] 0.1

deltaAFLQ(a,b,A.RR,A.N,method="Kpsi")

## [1] 0.13

deltaAFLQ(a,b,A.RR,A.N,method="otro")

## [1] 0.11
```

Integramos la función de selección

Actualización del Marco Input-Output

Metodos de Proyeccion de Marcos Input-Output

El Manual de EUROSTAT (2008a) dedica un capítulo completo a estos métodos diferenciando tres grupos:

- Métodos Univariantes, son aquellos que obtienen la matriz proyectada a partir de una corrección de la matriz de referencia, bien por filas, o bien por columnas, de acuerdo con una matriz diagonal de factores de corrección.
- Métodos Biproporcionales, a diferencia de los anteriores, la matriz proyectada es obtenida a través de una modificación tanto por filas como por columnas de la matriz de referencia.
- Métodos Estocásticos, son aquellos que toman en consideración a multitud de variables que ejercen su influencia sobre los elementos de la matriz de referencia para obtener la matriz proyectada.

Métodos basados en técnicas biproporcionales:

El Método RAS

El método RAS es el método más fructífero y que mayor número de aplicaciones ha tenido en el campo de las proyecciones de tablas Input-Output. El método originalmente fue desarrollado para la proyección de la matriz de coeficientes técnicos, aunque posteriormente su uso fue generalizado para matrices de transacciones como las que representan los elementos de la tabla de Origen y la tabla de Destino (Valderas, 2015).

El método RAS es una técnica de ajuste automático de una matriz por filas y columnas habitualmente utilizado como método de actualización temporal de matrices Input-Output. La técnica RAS se basa en un proceso de cálculo que puede considerarse en grandes líneas como la resolución de un problema estadístico de ajuste de una matriz desfasada temporalmente para que concuerde con los nuevos datos de la contabilidad nacional o regional, normalmente disponibles con periodicidad anual.

El método RAS básico fue desarrollado en el Departamento de Economía Aplicada de la Universidad de Cambridge (Reino Unido), por el premio Nobel Richard Stone en los primeros años de la década de los sesenta. Se trata de la aproximación más robusta desde una perspectiva teórica (Bacharach, 1970), pero presenta la limitación de que hay que conocer los marginales de la matriz que se quiere estimar.

Este método establece un procedimiento de cálculo por el cual una matriz de coeficientes va a ser corregida sucesivamente por factores correctores de fila (R) y de columna (S). Matemáticamente, el método vendría expresado a través de la operación matricial que le da el nombre:

$A_1 = RxA_0xS$

Donde,

A_0 es la Matriz original.

A_1 es la Matriz estimada.

R y S son matrices diagonales, que premultiplicando y post multiplicando, respectivamente, a la matriz de partida (A_0) proporcionan la convergencia de las filas y columnas agregadas de la matriz estimada (A_1).

Es un método iterativo en donde los factores de corrección se obtienen en la primera iteración dividiendo los marginales de la matriz estimada por los de la matriz original, y en las sucesivas dividiendo los marginales de la matriz estimada por los que la matriz obtenida en cada iteración, hasta alcanzar un determinado grado de convergencia o diferencia decimal entre unos y otros marginales.

Las aplicaciones del método RAS aplicado a las proyecciones de tablas Input-Output han sido innumerables desde los primeros trabajos realizados con el mismo por Stone y Brown. Para un mayor detalle de los orígenes de este método, sus primeras aplicaciones al campo Input-Output: Bacharach (1970), Lecomber (1975), Polenske (1997) y Miller y Blair (2009).

El principal inconveniente del método RAS original, es que requiere que los totales por filas y por columnas sean conocidos a priori, situación que en la práctica no se da, en especial los totales por filas que corresponden al output por productos, por lo que estos deberían ser estimados por algún otro procedimiento previamente para poder aplicar el método RAS. Temurshoev y Timmer (2011) han desarrollado una versión del RAS para trabajar con un marco Input-Output integrado que supera este problema permitiendo la estimación endógena de los elementos desconocidos de las marginales de la tabla a proyectar. En Valderas (2015) se exponen las diferentes extensiones del metodo original RAS que han tratado de dar respuesta a los incovenientes que presenta el modelo.

El Método MEURO

Además de la familia de métodos RAS y sus extensiones , existen otro tipo de métodos biproporcionales que también se han empleado como técnicas de proyección de elementos de un marco Input-Outpu, entre los que destaca el método EURO, desarrollado originalmente por Beutel (2002) para la actualización de tablas Input-Output simétricas y, posteriormente adaptado por el propio Beutel (2008) para su empleo en la actualización de tablas de Origen y Destino.

La idea básica del método EURO (Beutel, 2002) es aprovechar las predicciones macroeconómicas oficiales que se estiman en el ámbito de las Contabilidades Nacionales o Regionales, y que se publican con periodicidad anual para, a partir de una tabla Input-Output de referencia, derivar una nueva tabla Input-Output proyectada para aquellos años en los que disponemos de dichas predicciones macroeconómicas y que, esta proyección sea a su vez consistente con las citadas predicciones macroeconómicas oficiales.

La versión del método EURO que aparece en Eurostat (2008) tiene como obtetivo la proyección de tablas Input-Output simétricas, y al igual que el RAS es un procedimiento iterativo. Pero a diferencia de aquel en el que sólo se actualiza la matriz de coeficientes técnicos, en el método EURO intervienen todos los elementos de la tabla simétrica:

- Consumos intermedios diferenciando el origen entre consumos interiores e importados
- Valor añadido y sus componentes
- Producción interior
- Demanda final diferenciada por usos y por origen de los empleos

El punto de inicio del procedimiento de iteración es una tabla Input-Output, que consta de seis cuadrantes para producción nacional, las importaciones y el valor añadido. El procedimiento de iteración comienza con la suposición que, en la primera iteración, las tasas de crecimiento propuestas para el valor añadido son el punto de partida de las tasas de crecimiento desconocidas que caracterizan los diferentes niveles de producción, consumos intermedios, y demandas finales sectoriales.

Table 4.1.3.1: Esquema de la tabla Input-Output para la proyección en el método EURO.

	Ramas homogeneas	Componentes de la demanda final	Total
Productos interiores	Cuadrante I	Cuadrante II	Producción interior
Productos importados	Cuadrante III	Cuadrante IV	Importaciones
VAB y sus componentes	Cuadrante V	Cuadrante VI	Valor añadido
Total	Producción interior	Demanda final	

Fuente: (Valderas, 2015)

Estas tasas de crecimiento de partida cambiarán ligeramente hasta que se alcancen los crecimientos de las variables exógenas proyectadas. El procedimiento iterativo se inicia proyectando los inputs empleados de bienes interiores e importados a partir de las tasas de los valores añadidos, y los output intermedios obtenidos con las tasas de crecimiento de los valores añadidos de cada sector, en tanto que los finales se proyectan con sus respectivas tasas de crecimiento. Se obtiene una medida ponderada para cada elemento de los tres cuadrantes que una vez agregada en terminos Input-

Output, ofrece una solución que no asegura el equilibrio contable de un marco Input-Output, por lo que se requiere de otra secuencia en donde partiendo de la tecnología derivada de la nueva situación se restaure el equilibrio Input-Output y se obtenga una tabla equilibrada.

Paso 1. Actualización de los inputs finales e intermedios.

Todas las transacciones de los cuadrantes I a IV son ponderadas con la media aritmética correspondiente a las tasa de crecimiento de los outputs (z) y de los inputs (s).

(1) $T2 = Z * T1$

(2) $T3 = T1 * S$

(3) $T4 = (T2 + T3)/2$ Media aritmética

(4) $T4 = \sqrt{(T1 * T2)}$ Media geométrica

donde,

$T1$ = matriz de consumos intermedios y demanda final de bienes y servicios (r x p)

$T2$ = matriz de transacciones obtenida a partir de las tasas de crecimiento de los outputs (r x p)

$T3$ = matriz de transacciones obtenida a partir de las tasas de crecimiento de los inputs (r x p)

$T4$ = matriz de transacciones para los cuadrantes I al IV (r x p)

Z = matriz diagonal de los crecimientos de los inputs, obtenida a partir de los crecimientos de los valores añadidos tanto para producciones interiores como para los bienes importados (r x r)

S = matriz diagonal de tasas de crecimiento de la producción (output) obtenida a partir de los crecimientos de los valores añadidos, del consumo final y las exportaciones por bienes (p x p)

r = numero de productos interiores e importados

p = numero de actividades (producción y demanda final)

Paso 2. Actualización de los valores añadidos por sector.

El valor añadido de cada sector se actualiza multiplicando el valor añadido del año base por la matriz diagonal de los crecimientos de los inputs. (w_i).

(5) $T5 = v * w_i$

donde,

$T5$ = vector fila de los valores añadidos obtenidos con las tasas de crecimiento de los inputs (1 x p)

v = vector de valores añadidos por sectores (1 x p)

Paso 3. Composición de la tabla Input-Output Matriz A.

Una primera aproximación a la tabla Input-Output actualizada se obtiene con los resultados de los pasos uno y dos, pero sin que se garantice el equilibrio Input-Output.

Paso 4. Cálculo de los coeficientes Input-Output.

En el paso 4, se asume la tecnología contenida la estructura de producciones interindustriales de la tabla A, calculándose las producciones interiores, las importaciones y valores añadidos derivados de los coeficientes técnicos de la tabla A.

(6) $a_{ij} = \frac{x_{ij}}{x-j}$

(7) $b_{ij} = \frac{m_{ij}}{x_j}$

(8) $c_j = \frac{v_j}{x_j}$

donde,

a_{ij} = coeficientes técnicos de los inputs interiores

b_{ij} = coeficientes técnicos de los inputs importados

c_j = Tasas de los valores añadidos

x_{ij}= consumos intermedios de los bienes y servicios interiores

m_{ij} = consumos intermedios de los bienes y servicios importados

v_j = valores añadidos

x_j = producción interior

Paso 5. Elaboración del modelo Input-Output. Utilizando los coeficientes del paso 4, se calcula la inversa de Leontief, que al ser multiplicada por el vector de la demanda final da como solución el vector de producción de la tabla A.

(9) $x = (I - A)^{-1}y$

donde,

X = vector del output (producción interior)

A = matriz de coeficientes aij

I = matriz identidad

$(I - A)^{-1}$ = matriz inversa de Leontief

y = vector columna de las demandas finales de la tabla A

Paso 6. Determinar las necesidades de inputs interiores e importados.

Los inputs primarios e intermedios son calculados balanceando la tabla Input-Output, según los siguientes pasos.

(10) $Z = B(I - A)^{-1}y$

donde,

B = matriz de coeficientes intermedios interiores, importados y valores añadidos

Z = matriz de necesidades de input

Paso 7. Composición de la tabla Input-Output.

Se compone una nueva tabla consistente, tabla B, en donde los valores añadidos y demandas finales no corresponden con los valores de partida. Estos se obtendrán a través de iteraciones sucesivas.

Paso 8. Iteración.

Las tasas de crecimiento del output (w_o) e input (w_i) son modificadas durante el proceso de iteración hasta obtener los valores de partida, los derivados del cuadro macroeconómico inicial. De tal forma que resulte una tabla B cuyos valores añadidos, importaciones y demandas finales permitan rescatar las tasas de crecimiento iniciales. El proceso se detiene en la iteración k que garantiza un nivel α de margen de error entre las tasas de crecimiento obtenidas por el modelo y las del cuadro macroeconómico ($dev(i) < \alpha$).

La desviación entre variables macroeconómicas a proyectar (las de partida) y las que se obtienen en el modelo es:

(11) $dev = \frac{pro}{mod}$

donde,

dev = desviacion

pro = variables macroeconómicas exogenas a proyectar

mod = proyecciones Input-Output (resultados del modelo)

Las desviaciones observadas se utilizan para corregir las tasas de crecimiento z y s en un procedimiento aditivo. En cuyo caso el multiplicador y la función de ajuste tipo A se definen como:

si $dev > 0$

(12) $mult = dev - I$

(13) $Z = Z + mult$

(14) $S = S + mult$

si $dev < 0$

(15) $mult = I - dev$

(16) $Z = Z - mult$

(17) $S = Z - mult$

donde,

$mult$ = matriz diagonal de multiplicadores de ajuste para las tasas de crecimiento

dev = matriz diagonal de factores de desviación

I = matriz identidad

Z = matriz diagonal de las tasas de crecimiento de la producción interior e importada

S = matriz diagonal de las tasas de crecimiento de la producción interior y demandas finales

La funcion de ajuste A es eficiente para encontrar una solución sin demasiadas iteraciones pero las fluctuaciones cíclicas pueden originar un sistema inestable. Por ello una función convexa tipo B es la recomendada para ajustar las tasas de crecimiento durante el proceso de iteración. Si el modelo subestima (o sobreestima) la variable macroeconómica a proyectar, las correspondientes tasas de crecimiento w_o y w_i , son incrementadas (decrementadas) respectivamente, de acuerdo a una función de ajuste convexa.

En este otro tipo de ajuste la function es definida como:

si $dev > 0$

(18) $mult = 1 + \frac{[(dev-1)100]^c}{100}$ para $dev > 0$

(19) $w_o = w_o * mult$

(20) $w_i = w_i * mult$

si $dev < 0$

(21) $mult = 1 - \frac{[(1-dev)100]^c}{100}$ para $dev < 0$

(22) $w_o = w_o * mult$

(23) $w_i = w_i * mult$

siendo c = elasticidad del ajuste

El Método EURO fue adoptado por EUROSTAT y con él se ha llevado a cabo la proyección de las tablas Input-Output de la mayoría de los países miembros, para aquellos años intermedios en los que los países no calculan las mismas. Todos los contrastes llevados a cabo por EUROSTAT apuntan a la utilidad del método para proyectar tablas Input-Output elaboradas siguiendo la metodología SEC, especialmente en aquellas situaciones en las que deba elaborarse una tabla Input-Output con unos requisitos de información bastante reducidos.

La función mteuro tiene como parámetros

T1.- Tabla base de destino interior e importaciones

act.- Indices de valores a proyectar

nd .- Número de sectores de demanda final

f.- Factor para corregir las desviaciones típicas

m.- número de iteraciones

Programación en R del método Euro

```
mteuro <- function (T1,act,nd,f,m) {
n <- dim(T1)[2]-nd
T1 <- as.matrix(T1)
act <- as.matrix(act)
# primera iteracción
wo<- c(act[1:n],act[1:n],act[n+nd+1]) # tasas 1 a n y tasa va
rm <- diag(wo)
T2 <- rm %*% T1
wi <- c(act[1:n],act[(n+1):(n+nd)]) # tasa 1 a n y tasa cons y export
cm <- diag(wi)
T3 <- T1 %*% cm
T4 <- (T3+T2)/2
T4 <- rbind(T4[1:(2*n),],T3[(2*n+1),])



# Saca (r1:r3, c1:c3) Cear funcion inversa de leotief

coef=t(T4)/colSums(T4)
coef2=t(coef)
coef2[is.nan(coef2)]<-0
coef4<-coef2[1:n,1:n] # n sectores
i <- c(rep(1,n))
Id <- diag(i)
leontief=Id-coef4
inversa=solve(leontief)
inversa[is.nan(inversa)]<-0
Df <- rowSums(T4[,(n+1):(n+nd)])
Dfinal <- matrix(Df[1:n],ncol=1)
O <- inversa %*% Dfinal
```

```r
O2 <- colSums(T4)
O3 <- c(0,O2[(n+1):(n+nd)])
dou <- diag(O3)
MIO2 <- coef2 %*% dou


# iteracciones
for (i in 1:m) {
sum1 <- colSums(T1)
sum2 <- colSums(MIO2)
sum3 <- rowSums(T1)
sum4 <- rowSums(MIO2)
pro <- c(MIO2[(2*n+1),1:n]/T1[(2*n+1),1:n],sum2[(n+1):(n+nd)]/sum1[(n+1):
(n+nd)],sum4[(2*n+1)]/sum3[(2*n+1)],sum(sum4[(n+1):(n+n)])/sum(sum3[(n+1)
:(n+n)]))
pro[is.nan(pro)]<-0
desv <- act/pro
desv[is.nan(desv)]<-1
delta <- desv-1
coef <- ifelse(delta<0,1-(((1-desv)*100)^f)/100,1+(((desv-1)*100)^f)/100)
rev1 <- c(coef[1:n],rep(coef[(n+nd+2)],n),coef[(n+nd+1)])
wo <- rev1*wo
rm <- diag(wo)
IOW1 <- rm %*% T1
rev2 <- c(coef[1:(n+nd)])
wi <- wi*rev2
cm <- diag(wi)
IOW2 <- T1 %*% cm
IOW3 <- 0.5*IOW2+0.5*IOW1
IOW4 <- rbind(IOW3[1:(2*n),],IOW2[(2*n+1),])
IOW4[is.nan(IOW4)]<-0
coef=t(IOW4)/colSums(IOW4)
coef2=t(coef)
coef2[is.nan(coef2)]<-0
coef4<-coef2[1:n,1:n]
i <- c(rep(1,n))
Id <- diag(i)
leontief=Id-coef4
inversa=solve(leontief)
inversa[is.nan(inversa)]<-0
Df <- rowSums(IOW4[,(n+1):(n+nd)])
Dfinal <- matrix(Df[1:n], ncol=1)
O <- inversa %*% Dfinal
O2 <- colSums(IOW4)
O3 <- c(0,O2[(n+1):(n+nd)])
dOu <- diag(O3)
MIO2 <- coef2 %*% dOu}
act=matrix(act,ncol=1)
pro=matrix(pro,ncol=1)
desv=matrix(desv,ncol=1)
```

```
resultados=data.frame(cbind(act,pro,desv))
names(resultados)=c("act","pro","desv")
list(resultados=resultados,MIO=as.matrix(MIO2))
}
```

El Método MEURO aplicado a Tablas de Orígen y Destino

El método EURO para la proyección de tablas de Origen y Destino (SUT-EURO) es menos conocido en la literatura sobre métodos de proyección, ya que por cuestiones de tiempo no alcanzó a ser publicado en el manual de EUROSTAT de 2008 (Valderas, 2015) y no fue hasta 2010 cuando se hace público en un documento de trabajo del Proyecto World Input-Output Database (Temurshoev y Timmer,2011).

Partiendo de la siguiente representación de las tablas de Origen y Destino:

Table 4.1.4.1: Tabla de Origen

	Ramas	Importación	Oferta
Productos	V'	m	q
$\sum$	$x'1$	M	

Fuente: (Valderas, 2015)

Table 4.1.4.2: Tabla de Destino

	Ramas	Demanda Final	Empleos
Productos	U	Y	q
Valor Añadido	W	w	
$\sum$	$x'1$	y	

Fuente: (Valderas, 2015)

Donde,

- V representa la tabla de Origen (producto por rama)
- m representa el vector columna de importaciones
- M es el total de importaciones de una economía
- q representa el vector de oferta total de productos
- x es un vector fila con la producción tal de cada rama
- U representa la tabla de Destino con los empleos intermedios (producto-rama)
- Y representa la matriz de demanda final, con los empleos finales de cada producto (producto-categorías de la demanda final)

- y representa el vector de totales de la demanda final por categoría
- W representa la matriz de componentes del valor añadido para cada rama (componentes del VAB-rama)
- w representa el vector columna con los totales de cada una de las componentes en que dividamos el VAB

Los requerimientos de información para la implementación del método SUT-EURO son los siguientes:

1) Las tablas de Origen y Destino del año de referencia, al que denominaremos 0, valoradas a precios básicos, y distinguiendo entre productos interiores y productos importados:

i) Sea $U_{b,0}^{d}$,y $U_{b,0}^{m}$ la tabla de Destino para la demanda intermedia de dimensión p x r, a precios básicos (d = origen interior, m = origen importado).

ii) Sea $Y_{b,0}^{d}$,y $Y_{b,0}^{m}$ la tabla de Destino para la demanda final de dimensión p x f , a precios básicos (d = origen interior, m = origen importado).

iii) Sea $V_{b,0}$ la traspuesta de la tabla de Origen (make-matrix en la terminología americana) con dimensión r x p, y m_0 el vector de importaciones por productos de dimensión p x 1.

iv) Sea v_0 el vector de valores añadidos por rama con dimensión r x 1.

2) Para el año de proyección, en el caso del método EURO, los requisitos de información del son:

i) Tasa de crecimiento del valor añadido bruto de cada rama entre el año base y el año t. Esta información está recogida en el vector g_t^{v} que contendrá los factores de crecimientos del VAB de cada rama.

ii) Tasa de crecimiento de cada una de las componentes de la demanda final entre el año base y el año t . Esta información está recogida en el vector g_t^{y} de tasas de crecimientos de la demanda final para cada componente f.

iii) Tasa de crecimiento del total de importaciones entre el año base y el año t . Esta información está recogida en el vector g_t^{m} de tasas de crecimientos de los totales de las importaciones con dimensión m x 1.

Paso 1. Obtención de la cuota de mercado por productos de la economía.

La matriz de coeficients de mercado del año base biene dada por:

$$D_0 = V_{b,0}(q_{b,0}^{\hat{d}})^{-1}$$

Cada elemento, $D_0(i,j)$, de la matriz es la proporción de la producción interior del producto j que es producido por la rama i.

El método SUT-EURO asume que esta cuota de mercado permanecerá constante a lo largo de la proyección, es decir, que ante un incremento de la oferta del producto j, todas las industrias responderán en la misma medida con respecto a su cuota de mercado para satisfacer dicho incremento.

Paso 2. Actualización de las tablas de Origen y Destino.

En el segundo paso se obtiene una primera proyección de las tablas de Origen y de Destino para el año t a partir de las tablas de Origen y de Destino del año de referencia 0. Esta primera proyección se realiza de manera similar al Método EURO aplicado a la tabla simétrica.

En primer lugar, se obtien los elementos de las tablas de Destino, en la parte correspondiente a la demanda intermedia, tanto interior como importada, que se actualizan por filas (productos) y columnas (ramas) de acuerdo con el crecimiento del VAB entre el año t y el año 0, g_t^v.

$$U_{t(1)}^d = \frac{1}{2}(\hat{g_t^v}\, U_{b,0}^d + U_{b,0}^d\, \hat{g_t^v})$$

$$U_{t(1)}^m = \frac{1}{2}(\hat{g_t^v}\, U_{b,0}^m + U_{b,0}^m\, \hat{g_t^v})$$

Esta transformación para actualizar la demanda intermedia es equivalente a la llevada a cabo en los cuadrantes I y III la matriz simétrica en el método EURO.

De manera análoga, obtenemos una primera versión de los elementos correspondiente a la demanda final.

$$Y_{t(1)}^d = \frac{1}{2}(\hat{g_t^v}\, Y_{b,0}^d + Y_{b,0}^d\, \hat{g_t^v})$$

$$Y_{t(1)}^m = \frac{1}{2}(\hat{g_t^v}\, Y_{b,0}^m + Y_{b,0}^m\, \hat{g_t^v})$$

El vector de VAB se actualiza con sus tasas de crecimiento, obteniendose, en consecuencia, un VAB proyectado del año que coincidirá con la estimación oficial:

$$v_{t(1)} = \hat{g_t^v}\, v_0$$

y la matriz de producción del nuevo año:

$$V_{t(1)} = D_0((U_{t(1)}^d l + Y_{t(1)}^d l)$$

Siendo l un vector de unos con la dimensión adecuada en cada caso.

Las ecuaciones matriciales descritas requieren trabajar con el mismo número de industrias que de productos, de no ser así alguno de los productos matriciales no serían posibles, por ejemplo $\hat{g_t^v}\, U_{b,0}^d$

Paso 3. Obtención de la oferta total consistente.

Las tablas de Origen y de Destino resultantes del paso 2 no están equilibradas. El valor de la producción interior por ramas que se calcula a partir de la tabla de Origen no será coincidente con el que se obtiene a partir de la tabla de Destino:

$$x_{out,t(1)} = l'V_{t(1)}{}' \neq l'(U^d_{t(1)} + U^m_{t(1)}) + v_{t(1)}{}' = x_{inp,t(1)}{}'$$

Para lograr que las tablas de Origen y Destino proyectadas sean consistentes, el SUT-EURO va a suponer que las estructuras productivas obtenidas en la tabla de Destino son correctas tanto para la parte interior como para la parte importada, que la demanda final total de cada componente en que desagregamos la demanda final también es válida, y de manera implícita que la composición estructural de cada componente de la demanda final también es correcta.

Estas estructuras se obtienen a través de las siguientes ecuaciones:

$$B^d_{t(1)} = U^d_{t(1)}\, \hat{x}_{inp,t(1)}{}'^{-1}$$

$$B^m_{t(1)} = U^m_{t(1)}\, \hat{x}_{inp,t(1)}{}'^{-1}$$

Y el vector de demanda final total se obtiene como:

$$y^d_{t(1)} = Y^d_{t(1)} l$$

Partiendo de estas hipótesis se obtiene un vector de producción por ramas :

$$x_{t(2)} = (I - D_0 B^d_{t(1)})^{-1} D_0 y^d_{t(1)}$$

Paso 4. Derivación de las nuevas tablas de Origen y Destino consistentes.

A partir del vector de producción por ramas, reconstruimos las nuevas matrices de demanda intermedia interior e importada.

$$U^d_{t(2)} = B^d_{t(1)}\, \hat{x}_{inp,t(2)}$$

$$U^m_{t(2)} = B^m_{t(1)}\, \hat{x}_{inp,t(2)}$$

La demanda final es coincidente con la obtenida tras la primera transformación:

$$Y^d_{t(2)} = Y^d_{t(1)}$$

$$Y^m_{t(2)} = Y^m_{t(1)}$$

Y a partir de estas magnitudes, reconstruimos el vector de VAB como saldo:

$$v_{t(2)}{}' = x_{t(2)}{}' - l'(U^d_{t(1)} + U^m_{t(1)})$$

Y la tabla de Origen manteniendo fija la cuota de mercado del año base:

$$V_{t(2)} = D_0(U^d_{t(2)} l + Y_{t(1)} dl)$$

De este modo, tras este segundo conjunto de transformaciones las tablas de Origen y Destino están equilibradas tanto desde el punto de vista de la producción por ramas, como desde el punto de vista de la oferta y demanda total de productos.

Paso 5. Comprobación de convergencia y calibración de multiplicadores.

Una vez que hemos reconstruido las tablas de Origen y Destino de manera consistente, hemos de comparar las desviaciones que existirán entre los valores oficiales que nos proporciona la Contabilidad Nacional con respecto a los derivados del método SUT-EURO. Para ello, definimos de manera similar al método EURO aplicado a la tabla simétrica:

$$dev = \frac{pro}{mod}$$

Y a partir de aquí definimos los factores correctores de manera análoga al al método EURO aplicado a la tabla simétrica:

Para $dev > 1$

$$corr(i) = 1 + \frac{[(dev(i) - 1)100]^c}{100} \forall i = 1, \dots, r + f + m$$

y para $dev < 1$

$$corr(i) = 1 - \frac{[(1 - dev(i))100]^c}{100} \forall i = 1, \dots, r + f + m$$

A partir de estos factores de corrección se definen los nuevos vectores multiplicador:

$$g_{t,1}^{d}(i) = corr(i) g_t^{v}(i) \forall i = 1, \dots, r$$

$$g_{t,1}^{y}(i) = corr(i) g_t^{y}(i) \forall i = 1, \dots, f$$

$$g_{t,1}^{m}(i) = corr(i) g_t^{v}(i) \forall i = 1, \dots p^m$$

El primer conjunto de multiplicadores se aplican al crecimiento del VAB de cada rama; el segundo conjunto de multiplicadores son los que se aplican a las componentes de la demanda final; y el último está formado por los factores que se aplicarán al crecimiento de las importaciones.

Paso 6. Actualización de las tablas de Origen y Destino, y comienzo de una nueva iteración.

Una vez definidos estos multiplicadores, se realiza una actualización de las tablas de Origen y Destino de manera similar a la llevada a cabo en el paso 2 del método pero con los nuevos multiplicadores definidos:

$$U_{t(3)}^{d} = \frac{1}{2} \left(g_{t(\hat{1,1})^d} \, U_{b,0}^{d} + U_{b,0}^{d} \, g_{t(\hat{1,1})^d} \right)$$

$$U_{t(3)}^{m} = \frac{1}{2}(g_{t(\hat{1,1})^m}\, U_{b,0}^{d} + U_{b,0}^{m}\, g_{t(\hat{1,1})^m})$$

$$Y_{t(3)}^{d} = \frac{1}{2}(g_{t(\hat{1,1})^d}\, Y_{b,0}^{d} + Y_{b,0}^{d}\, g_{t(\hat{1,1})^y})$$

$$Y_{t(3)}^{m} = \frac{1}{2}(g_{t(\hat{1,1})^m}\, Y_{b,0}^{d} + Y_{b,0}^{m}\, g_{t(\hat{1,1})^y})$$

$$v_{t(3)} = g_{t(\hat{1,1})^d}\, v_0$$

$$V_{t(3)} = D_0(U_{t(3)}^{d} l + Y_{t(3)} dl$$

Al igual que ocurría en el paso 2, las nuevas tablas de Origen y de Destino no nos aseguran la igualdad de outputs e inputs por rama. Por tanto, a partir de este punto daríamos comienzo a una nueva iteración repitiendo los pasos 3, 4, 5 y 6 de nuevo. Como resultado de todo el proceso, se obtendrá un nuevo vector de desviaciones que cuantifica la diferencia que habrá entre las estimaciones oficiales y las proyectadas por las nuevas tablas de Origen y Destino. El procedimiento continuará hasta alcanzar la convergencia, esto es, si las desviaciones son aceptables, digamos inferiores a un nivel para $dev(i) < \alpha$ prefijado el procedimiento se da por terminado.

El Método SUT-RAS

El método SUT-RAS ofrece un método de proyección integrado, de manera similar al SUT-EURO,en el que todas las componentes de las tablas de Origen y Destino se estiman de manera conjunta. El método SUT-RAS fue desarrollado por Temurshoev y Timmer (2011) para el proyecto de las tablas Input-Output mundiales (WIOD).

El método SUT-RAS ofrece un método de proyección integrado, de manera similar al SUT-EURO,en el que todas las componentes de las tablas de Origen y Destino se estiman de manera conjunta y no de manera separada.

Partiendo de la descripción de MIO que figura en las Tablas 8 y 9, las tablas de Origen y Destino se pueden representar de manera integrada en una única tabla, que denominamos marco integrado, del siguiente modo:

Table 4.1.5.1: Marco Integrado

	Productos	Ramas	Demanda Final	∑
Productos		U	Y	q
Ramas	V			x
Valor Añadido		W		w
Importaciones	m’			M
∑	q’	x’	y	

Fuente: (Valderas, 2015)

En esta tabla están representadas de manera integrada tanto la tabla de Origen como la tabla de Destino, y en ella se producen una serie de identidades contables (prescindiendo de diferencias en los criterios de valoración):

$$U \cdot \iota + Y \cdot \iota = V' \cdot \iota + m = q$$

$$U' \cdot \iota + W' \cdot \iota = V \cdot \iota = x$$

La información del año de referencia 0 se encontraría resumida en el siguiente marco integrado:

Table 4.1.5.2: Marco Integrado

	Productos	Ramas	Demanda Final	$\sum$
Productos		$U_{b,0}$	$Y_{b,0}$	$q_{b,0}$
Ramas	$V_{b,0}$			$x_{b,0}$
Importaciones	m'_0			M_0
$\sum$	$q'_{b,0}$	$u'_{b,0} = x'_{b,0} - v'_{b,0}$	$y_{b,0}$	

Fuente: (Valderas, 2015)

Para el año que queremos proyectar las tablas de Origen y de Destino, necesitamos la siguiente información de referencia:

i) Vector con los valores añadidos brutos para cada rama, $v_{b,t}$.

ii) Vector con los valores totales de la demanda final diferenciada por componente $y_{b,t}$.

iii) Valor del total de importaciones para el año t ,M_t.

iv) Vector con los valores del total de producción por rama $x_{b,t}$.

Una vez definida la información del año de referencia, y de la información requerida para realizar la proyección del año t , se define la matriz A.

$$A = \begin{pmatrix} O_{p \cdot p} & \overline{U}_0 \\ \overline{V}_0 & O_{(r+m)\cdot(r+f)} \end{pmatrix}$$

donde $\overline{V}_0 = (V'_{b,0} + m_0)'$ es la tabla de Origen del año de referencia.

$\overline{U}_0 = (U_{b,0} + y_{b,0})$ es la tabla de Destino del año de referencia.

O es una tabla de ceros con las dimensiones adecuadas.

Nuestro objetivo es obtener una matriz X para el año t que contenga las tablas de Origen y Destino de dicho año, y que dicha matriz verifique la información adicional de la que disponemos obtenida a partir de fuentes oficiales:

$$X = \begin{pmatrix} O_{p\cdot p} & \overline{U}_t \\ \overline{V}_t & O_{(r+m)\cdot(r+f)} \end{pmatrix}$$

donde

$\overline{V}_t = (V'_{b,t} + m_t)'$ es la tabla de Origen proyectada.

$\overline{U}_t = (U_{b,t} + y_{b,t})$ es la tabla de Destino proyectada.

O es una tabla de ceros con las dimensiones adecuadas.

Denominemos de manera breve como a_{ij} a los elementos de la matriz A y x_{ij} a los elementos de la matriz X . Para lograr la proyección de X, el SUT-RAS va a plantear un problema de optimización restringida.

$$\min_{z_{ij}} \sum^{i} \sum^{j} \mid a_{ij} \mid (z_{ij} \frac{z_{ij}}{e} + 1)$$

donde $z_{ij} = \frac{x_{ij}}{x_{ij}}$, con $z_{ij} = 1$ si $a_{ij} = 0$.

sujeto a

$\sum^{j\epsilon(r,f)} a_{uj}\, z_{uj} - \sum^{i\epsilon(r,m)} a_{ui}\, z_{ui} = 0$ para todo $u\epsilon(p)$

$\sum^{i\epsilon(p)} a_{ij}\, z_{ij} = \overline{u}_j$ para todo $j\epsilon(r,f)$

$\sum^{j\epsilon(p)} a_{ij}\, z_{ij} = \overline{x}_i$ para todo $i\epsilon(r,m)$

Modelo de impacto con matriz orlada

La Tabla Input-Output simétrica se estructura en tres matrices independientes: la matriz de consumos intermedios, la matriz de demanda final y la matriz de inputs intermedios. La matriz de consumos intermedios contabiliza las relaciones de intercambio entre las distintas ramas productivas. La matriz de demanda final recoge la parte de la producción de bienes y servicios que se destina a los usuarios finales (demanda de consumo, demanda de inversión y demanda exterior de bienes producidos en la economía nacional). Y finalmente, la matriz de inputs primarios en donde se registran los pagos que realizan las empresas y las administraciones por utilizar los factores originarios de la producción (rentas del trabajo y excedentes empresariales). La matriz de inputs primarios proporciona el Valor Añadido de cada rama que se obtiene deduciendo del valor de la producción el total de consumos intermedios. Cada elemento x_{ij} de la matriz de consumos intermedios recoge los consumos de productos de la rama i que hace la rama j. Si estos consumos son originarios de empresas residentes en el área territorial de referencia de la tabla input-output, es decir, tienen el carácter de interior, se referencian con el superíndice r, los importados desde unidades no residentes se referencian con el superíndice m. La producción que realiza una rama (X_j) se obtiene como suma de los elementos que figuran en cada columna: consumos intermedios de unidades residentes, importaciones y valor añadido (V). Por filas, aparecen los destinos de la producción interior (X_i) y de las importaciones (M_i). Estos destinos son la demanda intermedia (las compras que realizan otros sectores) y la demanda final (D_i).

Dado el equilibrio contable de una TIO, en donde el valor de producción por columnas ha de igualarse con la producción distribuida o empleada en cada fila, se puede también representar la estructura formal de la TIO a través del siguiente sistema de ecuaciones lineales:

$$x_{11}^r + x_{12}^r + \ldots + x_{1n}^r + D_1^r = X_1$$

$$x_{21}^r + x_{22}^r + \ldots + x_{2n}^r + D_2^r = X_2$$

$$\ldots$$

$$x_{n1}^r + x_{n2}^r + \ldots + x_{nn}^r + D_n^r = X_n$$

Definimos el coeficiente técnico a_{ij} como la relación entre la cantidad consumida de un input y el valor de producción de una rama: $\frac{x_{ij}}{X_j}$

obteniendo un nuevo sistema de ecuaciones:

$$a_{11}^r X_1 + a_{12}^r X_2 + \ldots + a_{1n}^r X_n + D_1^r = X_1$$

$$a_{21}^r X_1 + a_{22}^r X_2 + \ldots + a_{2n}^r X_n + D_2^r = X_2$$

$$\ldots$$

$$a_{n1}^r X_1 + a_{n2}^r X_2 + \ldots + a_{nn}^r X_n + D_n^r = X_n$$

Este nuevo sistema de ecuaciones en notación matricial, queda expresado por:

$$A^r X + D^r = X$$

Operando convenientemente se transforma en:

$$D^r = (1 - A^r)X$$

En donde, I es la matriz Identidad y

$$X = (1 - A^r)^{-1} D^r$$

A la matriz $(1 - A^r)^{-1} D$ se la conoce como la matriz inversa de Leontief, cuyos elementos A_{ij}^r constituyen una medida del esfuerzo de producción requerido a la rama i por parte de la rama j para abastecer una unidad de demanda final de esta última.

Cada elemento de la matriz inversa de Leontief representa pues los efectos acumulativos (directos e indirectos) que subyacen en la estructura productiva que la TIO representa.

Consideremos, asimismo, un segundo nivel de endogenización de las variables en el contexto de las tablas Input-Output formado por el vector de Demanda Final (compuesto por las variables Consumo Privado, Consumo Público, Formación Bruta de Capital y Exportaciones). Si suponemos que el consumo realizado de los bienes y servicios producidos por un sector es una proporción constante del VAB total tenemos que:

$$C_i = k_i l'X$$

donde k_i es una constante que indica la proporción del VAB que se dedica al consumo de bienes y servicios producidos por el sector i-ésimo e l' es un vector cuyo elemento i-ésimo indica para cada sector la proporción que representa el VAB sobre la producción total X_i, de manera que el producto $l'X$ es el VAB agregado de la economia.

Es decir, si:

$k_i = \frac{C_i}{VAB}, l = (l_1, l_2, \ldots, l_n), l_i = \frac{VAB_i}{X_i}$

Por tanto, en notación matricial, quedaría:

$C = kl'X = KX$

A partir de la expresión anterior, el sistema de ecuaciones descrito ($A^r X + D^r = X$) se puede reformular ahora como:

$$A^r X + KX + D^* = X$$

siendo D* ahora el vector suma del Consumo Público, la Formación Bruta de Capital y las Exportaciones.

Operando en el modelo quedaría entonces:

$$X = (I - A - K)^{-1} D^* = (I - A^*)^{-1} D^*$$

pudiendo obtenerse las producciones sectoriales en función de la nueva variable exógena.

A partir del instrumental desarrollado pueden realizarse las predicciones con el modelo Input-Output, las cuales nos permitirán valorar los impactos o efectos sectoriales que tiene un aumento de la Demanda Final de un complejo industrial en el conjunto de la economía. Dichos efectos o impactos macroeconómicos se pueden dividir en tres tipos:

Un efecto directo provocado por el aumento de la demanda final del complejo el cual provoca un aumento de la producción del mismo con objeto de cubrir el aumento de demanda.

Unos efectos indirectos en el resto de sectores que suministran inputs a las ramas que forman el complejo, las cuales, ante el aumento de demanda, realizarán mayores pedidos a sus proveedores para poder aumentar su producción.

Unos efectos inducidos producidos a causa del aumento de demanda de inputs que realizan las ramas afectadas por los efectos indirectos, lo cual se transmite al conjunto de sectores de la economía.

Finalmente, los efectos señalados anteriormente producen a su vez un incremento de las rentas salariales lo que, dado el supuesto de consumo como variable dependiente de la renta, provoca un aumento del consumo lo que da como resultado nuevos aumentos de demanda final. Es lo que denominamos efecto renta.

Los elementos de la última fila de la nueva matriz, A^*, indican la renta doméstica directamente generada al obtener una unidad del sector j. La última columna de la nueva matriz representa las necesidades directas de producto i para la obtención de una unidad final de consumo privado.

$$B^* = (I - A^*)^{-1}$$

Los multiplicadores se calculan utilizando la última fila de la nueva matriz inversa de Leontief, B*. En forma de matriz particionada podemos expresar la nueva matriz de transacciones intersectoriales como:

$$\begin{bmatrix} \\ VAB \end{bmatrix} = \begin{bmatrix} & K \\ l' & 0 \end{bmatrix} \begin{bmatrix} \\ VAB \end{bmatrix} + \begin{bmatrix} Y - C \\ RE \end{bmatrix}$$

siendo RE las rentas recibidas por el exterior.

La matriz inversa de Leontief B^* es igual a:

$$B^* = \begin{bmatrix} & K \\ l' & 0 \end{bmatrix}^{-1}$$

El aumento de la demanda final del sector j tiene como efecto directo inmediato el aumento de la producción sectorial para satisfacerla. Es decir:

$\varDelta X_j = \varDelta D_j$

El segundo de los efectos (efecto indirecto) se deduce de los coeficientes técnicos de producción los cuales nos miden el consumo de mercancía necesaria del sector i para obtener una unidad del sector j tal que:

$\varDelta x_{ij} = a_{ij} \varDelta X_j$

Dado que el efecto total inicial viene determinado por la resolución del siguiente modelo matricial:

$$\varDelta X = (1 - A)^{-1} \varDelta D$$

podemos obtener el efecto inducido como la diferencia entre el efecto total inicial y los efectos directo e indirecto.

Por ultimo, los multiplicadores totales extendidos a los efectos inducidos por el aumento de la renta, que se obtienen a partir de:

$$\Delta X = B^{*}\Delta D$$

De manera que la diferencia entre el efecto total extendido y el efecto total inicial, daría como resultado los efectos renta:

$$EfectoTotal \\ = EfectoDirecto + EfectoIndirecto + EfectoInducido + EfectoRenta$$

La función MIOR = function(TO,TDu,TDy,w,vab,Empleo,vector), permite realizar estos cálculos, dando como resultado una lista en donde se recogen los diferentes efectos para el VAB por ramas de actividad del MIO, y en otra lista los del empleo.

La función solicita los datos de la matriz base:

TO sería la tabla de origen, con dimension n*n ramas.

TDu la tabla de destino de la producción interindustrial las $n * n$ primeras filas son las compras interiores x_{ij}^{r}, las $n * n$ siguientes filas son los compras importadas x_{ij}^{m}, y la fila final es la fila de los totales CI_j.

TDy la tabla de destino de las demanda final las $n * n$ primeras filas son las compras interiores d_{ij}^{r}, las $n * n$ siguientes filas son los compras importadas d_{ij}^{m}, y la fila final los totales de las partidas de la demanda final D_k.

W, es un vector fila con los coeficiente de los salarios sobre el VAB en el para las n ramas,se añade un cero en la ultima celda.

vab, es un vector fila con los valores añadidos de los n sectores VAB_j.

Empleo, es un vector fila con los trabajadores ocupados en cada rama E_j.

```
setwd("G:/actualizacion MIO")
#Toda la información sobre las tablas es la de la proyección Actualizació
n SUT-EURO-simetrica:
TO_t = read.csv("X_TO_t.csv",dec=".",header=TRUE,sep=",")[,-1]
TDu_t = read.csv("X_TDu_t.csv",dec=".",header=TRUE,sep=",")[,-1]
TDy_t = read.csv("X_TDy_t.csv",dec=".",header=TRUE,sep=",")[,-1]
proyvab=read.csv("proyvab.csv",dec=".",header=TRUE,sep=",")[,-1]#
# Distribución de los salarios sobre el VAB en el año cero para las r ram
as,se añade un cero en la ultima celda.
w=read.csv("w_0.csv",dec=",",header=FALSE,sep=";")
# Empleos total(primera fila) y asalariado (segunda fila) en el periodo t
Empleo=read.csv("empleo_t.csv",dec=",",header=FALSE,sep=";")
Empleo=Empleo[1,]
# Vector de demanda final a evaluar de p productos,y salarios finales de
```

```
la empresa en t.
vector=read.csv("vector_t.csv",dec=",",header=FALSE,sep=";")

#ojo los datos de la empresa están en miles y las tablas en euros
vector=vector*1000

MIOR = function(TO,TDu,TDy,w,vab,Empleo,vector){
## Coeficientes tecnicos interior e inversa de leontief
r=dim(TO)[2]-1
p=dim(TO)[1]
CI_int_t=as.matrix(TDu[1:r,1:r])
PT=colSums(TO)[1:r]
EVAB=matrix(vab/PT)
Prodt=matrix(as.numeric(Empleo)/PT)
EVAB[is.na(EVAB)]=0
Prodt[is.na(Prodt)]=0
PT=diag(1/PT)
PT[is.na(PT)]=0
CT=CI_int_t%*%PT
CT[is.na(CT)]=0
I=diag(c(rep(1,r)))
B=solve(I-CT)

## Matriz orlada, e inversa de leontief de la matriz orlada.

VC=TDy_t[1:r,1]/sum(TDy_t[1:r,1])
CTO=cbind(CT,VC)
CTO=rbind(CTO,as.matrix(w))
I=diag(c(rep(1,r+1)))
BO=solve(I-CTO)

## Multiplicadores I


vector_I=as.matrix(t(vector[1:p]))
Mult_I=B%*%vector_I


## Multiplicadores II


vector_II=as.matrix(t(vector))
Mult_II=BO%*%vector_II

## Efectos VAB

Indirectos_VAB=vector_I*EVAB
Inducidos_I_VAB=Mult_I*EVAB-Indirectos_VAB
Inducidos_II_VAB=Mult_II[1:p]*EVAB-Indirectos_VAB-Inducidos_I_VAB
```

```
Totales_VAB=Indirectos_VAB+Inducidos_I_VAB+Inducidos_II_VAB
Efectos_VAB=data.frame(Indirectos=Indirectos_VAB,Inducidos_I=Inducidos_I_
VAB,Inducidos_II=Inducidos_II_VAB,Totales=Totales_VAB)

## Efectos Empleo

Indirectos_Empleo=vector_I*Prodt
Inducidos_I_Empleo=Mult_I*Prodt-Indirectos_Empleo
Inducidos_II_Empleo=Mult_II[1:p]*Prodt-Indirectos_Empleo-Inducidos_I_Empl
eo
Totales_Empleo=Indirectos_Empleo+Inducidos_I_Empleo+Inducidos_II_Empleo
Efectos_Empleo=data.frame(Indirectos=Indirectos_Empleo,Inducidos_I=Induci
dos_I_Empleo,Inducidos_II=Inducidos_II_Empleo,Totales=Totales_Empleo)

list(VAB=Efectos_VAB,Empleo=Efectos_Empleo)
}
impacto= MIOR(TO_t,TDu_t,TDy_t,w,proyvab,Empleo,vector)

impacto

## $VAB
##        Indirectos  Inducidos_I Inducidos_II    Totales
## V1      1572.4108 4.918191e+04   132895.700  183650.02
## V2         0.0000 4.602471e+02    62917.025   63377.27
## V3         0.0000 1.966739e+04     4213.365   23880.76
## V4         0.0000 1.033580e+03   167810.210  168843.79
## V5         0.0000 4.773438e+02    59357.126   59834.47
## V6     13062.0350 6.857852e+03   342071.690  361991.58
## V7       240.7662 1.929076e+04   112047.990  131579.52
## V8    394508.6122 7.316657e+04    40442.584  508117.76
## V9         0.0000 1.307522e+04    57233.920   70309.14
## V10        0.0000 0.000000e+00        0.000       0.00
## V11   486020.2364 1.183545e+04   116005.944  613861.63
## V12      674.2060 1.684183e+04    21067.959   38583.99
## V13     2616.9253 1.929022e+04     8530.769   30437.91
## V14   411959.4014 3.300772e+04     6154.102  451121.22
## V15    15349.1440 3.317451e+03     1451.262   20117.86
## V16   273111.6719 7.550900e+04     4829.897  353450.57
## V17   731818.6429 5.001916e+04    58294.358  840132.16
## V18    13073.5553 8.782120e+03     7793.578   29649.25
## V19        0.0000 2.524448e+04     9149.073   34393.56
## V20    75318.8869 3.630497e+04    37718.994  149342.85
## V21        0.0000 1.509171e+04    17498.649   32590.36
## V22   145692.5044 7.376308e+04   135151.344  354606.93
## V23  2985928.3860 2.011300e+06   778332.925 5775560.92
## V24  1848018.2367 2.065144e+05   763365.803 2817898.44
## V25   753925.5590 3.639165e+05   316041.095 1433883.12
## V26   180285.0457 1.633135e+05   706236.164 1049834.74
## V27    32368.8319 4.018655e+05   609640.133 1043874.44
## V28    70882.3669 3.018940e+05  3989699.012 4362475.42
```

```
## V29 2684412.9308 7.001289e+05  1131438.675 4515980.48
## V30       0.0000 1.908440e+05   153504.443  344348.48
## V31     239.8994 2.535971e+04   538133.952  563733.56
## V32       0.0000 4.411796e+04  2274950.863 2319068.82
## V33    6250.3878 2.172015e+02    20974.110   27441.70
## V34    7989.9904 2.330149e+05  1075567.247 1316572.09
## V35  159043.8316 4.547288e+04    93633.653  298150.36
## V36   18783.0009 1.582675e+05   841668.472 1018718.98
## V37       0.0000 1.770124e+05   559367.390  736379.74
## V38       0.0000 1.983177e+05  7771053.302 7969371.01
## V39   11381.7389 1.941530e+05   231485.054  437019.76
## V40       0.0000 7.689928e+04   220602.196  297501.48
## V41       0.0000 7.465181e+04   319923.107  394574.92
## V42  108038.2127 9.455579e+04   126309.726  328903.73
## V43       0.0000 1.512208e+04    32169.579   47291.66
## V44  211967.9463 2.371017e+05   857495.510 1306565.16
## V45       0.0000 0.000000e+00  3405403.253 3405403.25
## V46  386474.0067 7.566612e+04  5893750.084 6355890.21
## V47       0.0000 4.719735e+03  4039980.642 4044700.38
## V48       0.0000 4.006937e+01   924684.429  924724.50
## V49    4789.1723 4.504726e+03   584551.722  593845.62
## V50       0.0000 4.091800e+03   253432.124  257523.92
## V51       0.0000 1.706490e+05  1187816.760 1358465.72
## V52       0.0000 0.000000e+00   532850.601  532850.60
##
## $Empleo
##       Indirectos  Inducidos_I Inducidos_II     Totales
## V1   0.012625237  0.394892558   1.06704926   1.4745671
## V2   0.000000000  0.022123192   3.02430035   3.0464235
## V3   0.000000000  0.139611935   0.02990920   0.1695211
## V4   0.000000000  0.018152818   2.94725943   2.9654123
## V5   0.000000000  0.013911734   1.72990744   1.7438192
## V6   0.123154846  0.064658964   3.22520848   3.4130223
## V7   0.005190211  0.415852182   2.41542529   2.8364677
## V8  12.070293081  2.238587189   1.23737181  15.5462521
## V9   0.000000000  0.270862974   1.18564311   1.4565061
## V10  0.000000000  0.000000000   0.00000000   0.0000000
## V11  2.620138905  0.063805011   0.62538896   3.3093329
## V12  0.006899527  0.172351833   0.21560021   0.3948516
## V13  0.059901995  0.441557316   0.19527117   0.6967305
## V14  6.003131635  0.480993194   0.08967846   6.5738033
## V15  0.436280227  0.094294405   0.04125032   0.5718249
## V16  4.660406858  1.288493665   0.08241788   6.0313184
## V17 10.563376450  0.721997503   0.84144515  12.1268191
## V18  0.208414109  0.140001534   0.12424254   0.4726582
## V19  0.000000000  0.152770897   0.05536703   0.2081379
## V20  1.046048539  0.504212947   0.52385132   2.0741128
## V21  0.000000000  0.167003414   0.19363833   0.3606417
## V22  2.506400090  1.268972590   2.32505674   6.1004294
## V23  8.062796173  5.431040776   2.10170470  15.5955417
```

```
## V24 57.748418445  6.453334652  23.85429265  88.0560457
## V25  8.159434067  3.938522026   3.42038607  15.5183422
## V26  5.234134869  4.741408501  20.50383777  30.4793811
## V27  0.602680195  7.482394257  11.35098218  19.4360566
## V28  1.706896927  7.269819568  96.07474012 105.0514566
## V29 35.231586057  9.188843587  14.84957050  59.2700001
## V30  0.000000000  2.022178482   1.62652911   3.6487076
## V31  0.005644826  0.596713195  12.66227609  13.2646341
## V32  0.000000000  1.513985405  78.06894374  79.5829292
## V33  0.094826369  0.003295224   0.31820405   0.4163256
## V34  0.017700636  0.516209761   2.38275928   2.9166697
## V35  4.379305723  1.252105294   2.57822256   8.2096336
## V36  0.177581535  1.496320343   7.95744941   9.6313513
## V37  0.000000000  1.286293976   4.06474967   5.3510436
## V38  0.000000000  0.174724734   6.84656583   7.0212906
## V39  0.265952104  4.536687217   5.40900977  10.2116491
## V40  0.000000000  4.923802320  14.12499072  19.0487930
## V41  0.000000000  1.845124895   7.90735095   9.7524758
## V42  1.849483478  1.618680653   2.16226968   5.6304338
## V43  0.000000000  0.527604381   1.12238581   1.6499902
## V44 10.623171835 11.882797067  42.97499838  65.4809673
## V45  0.000000000  0.000000000  59.72011204  59.7201120
## V46  8.046776961  1.575444603 122.71379591 132.3360175
## V47  0.000000000  0.087952507  75.28524642  75.3731989
## V48  0.000000000  0.001424639  32.87651649  32.8779411
## V49  0.088075555  0.082844433  10.75023278  10.9211528
## V50  0.000000000  0.131674097   8.15544366   8.2871178
## V51  0.000000000  2.514247282  17.50063400  20.0148813
## V52  0.000000000  0.000000000  66.57430876  66.5743088
```