

Introducción a deaR

Seminario informal

V. Coll, V. J. Bolós y R. Benítez

25 de enero de 2019

¿Qué es DEA?

DEA es una técnica de programación matemática que permite la construcción de una superficie envolvente, frontera eficiente o función de producción empírica, a partir de los datos disponibles del conjunto de DMUs objeto de estudio.

Las DMUs que determinan la envolvente son denominadas DMUs eficientes y aquellas que no permanecen sobre la misma son consideradas DMUs ineficientes.

DEA permite la evaluación de la eficiencia **relativa** de cada una de las DMUs.

Caracterización de los modelos DEA

- Tipología de medida de eficiencia: modelos radiales, modelos no radiales, etc.
- Orientación del modelo.
- Tipología de rendimientos a escala.

El paquete deaR

- Paquete de [R](#)
- Desarrollado con el IDE [Rstudio](#)
- No sólo es un conjunto de funciones que permite resolver los modelos DEA más usuales
- Proporciona una infraestructura de datos flexible que permite incorporar nuevos modelos de forma sencilla

Modelos disponibles

Modelos radiales

Modelo	Orientación	Retornos a escala	Función
Radial básico	IO/OO/DIR	CRS/VRS/NIRS/NDRS/GRS	model_basic
Multiplicativo	IO/OO	CRS/VRS/NIRS/NDRS/GRS	model_multiplier
Free disposal hull	IO/OO/DIR	VRS	model_fdh

Modelos disponibles

Modelos no radiales

Modelo	Orientación	Retornos a escala	Función
No radial básico	IO/OO	CRS/VRS/NIRS/NDRS/GRS	model_nonradial
Aditivo	NA	CRS/VRS/NIRS/NDRS/GRS	model_additive
Preference Structure	IO/OO	CRS/VRS/NIRS/NDRS/GRS	model_deaps
Slack-based model	IO/OO/NO	CRS/VRS/NIRS/NDRS/GRS	model_sbmeff
Eficiencia de beneficios ¹	NA	CRS/VRS/NIRS/NDRS/GRS	model_profit

¹: Sólo en la versión de desarrollo (github)

Modelos disponibles

Modelos de supereficiencia

Modelo	Orientación	Retornos a escala	Función
Superef. básica	IO/OO/DIR	CRS/VRS/NIRS/NDRS/GRS	model_supereff
Superef. basada en slacks	IO/OO/NO	CRS/VRS/NIRS/NDRS/GRS	model_sbmeff
Superef. aditiva	NA	CRS/VRS/NIRS/NDRS/GRS	model_addsupereff
Superef. compuesta ²	IO/OO/NO	CRS/VRS/NIRS/NDRS/GRS	model_sbmcomposite

²: Sólo en la versión de desarrollo (github)

Modelos disponibles

Modelos fuzzy

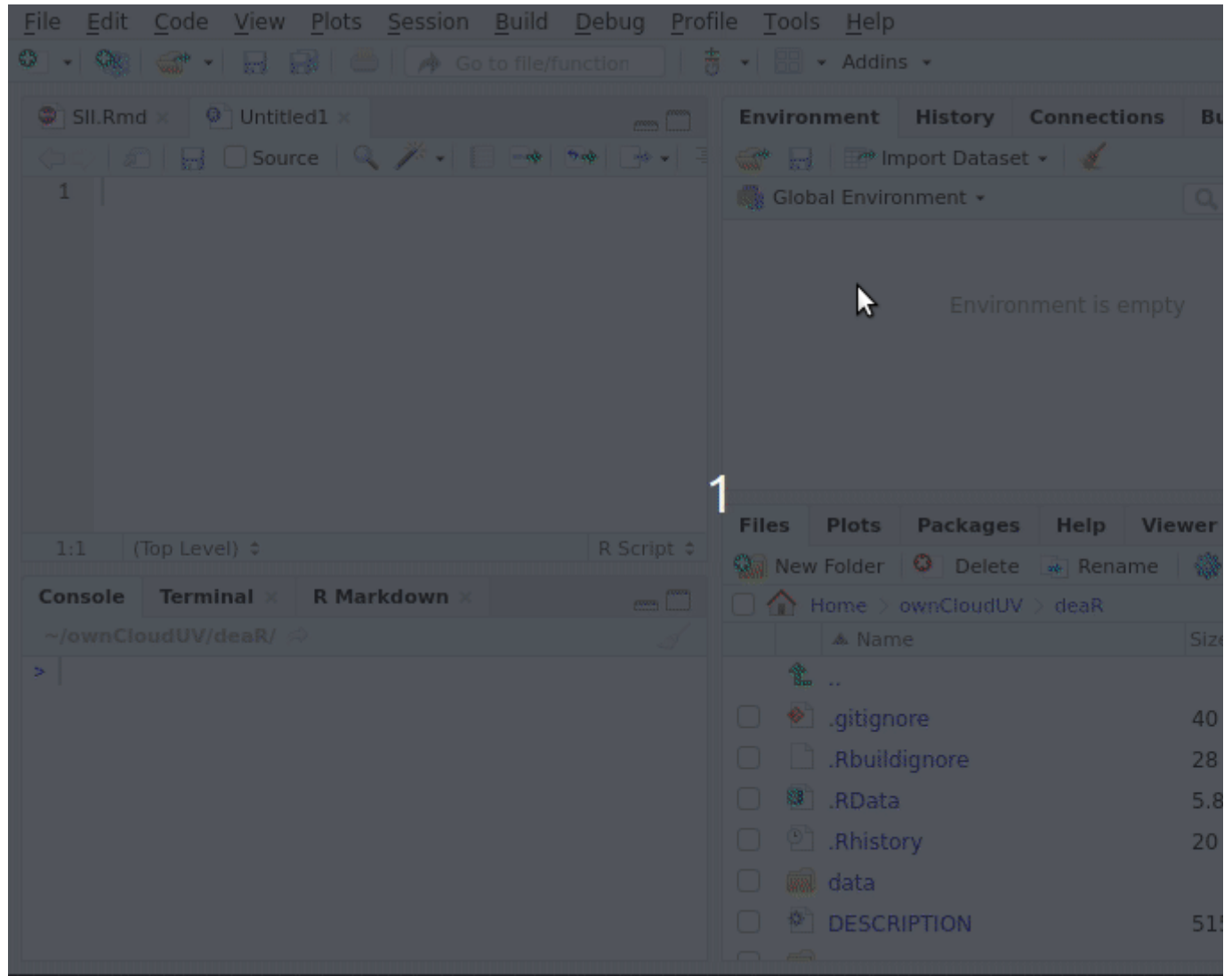
Modelo	Orientación	Retornos a escala	Función
Modelo Guo-Tanaka	IO/OO	CRS	modelfuzzy_guotanaka
Modelo Kao-Liu	Todos	Todos	modelfuzzy_kaoliu
Modelo Posibilístico	IO/OO/DIR	CRS/VRS/NIRS/NDRS/GRS	modelfuzzy_possibilisti

Modelos disponibles

Otros modelos

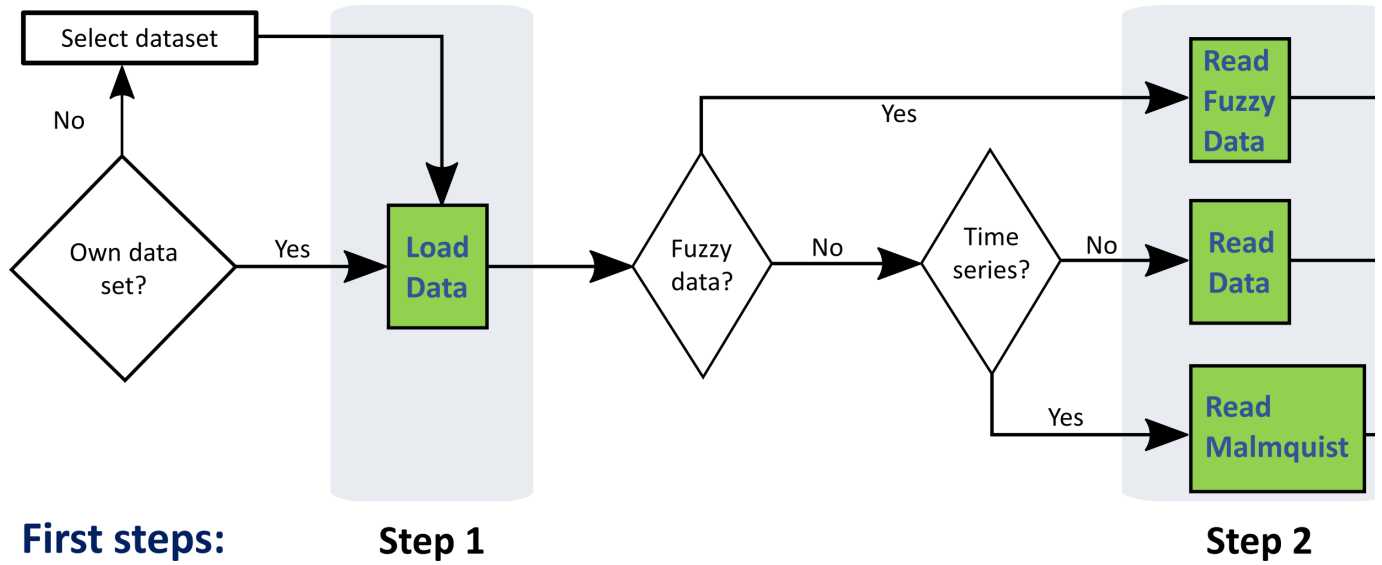
Modelo	Orientación	Retornos a escala	Función
Índice de Malmquist	IO/OO	VSR	malmquist_index
Bootstrap básico	IO/OO	CRS/VRS	bootstrap_basic
Eficiencias cruzadas	IO/OO	CRS/VRS	cross_efficiency
Eficiencias cruzadas fuzzy	IO/OO	CRS	cross_efficiency_fuzzy

Instalar deaR

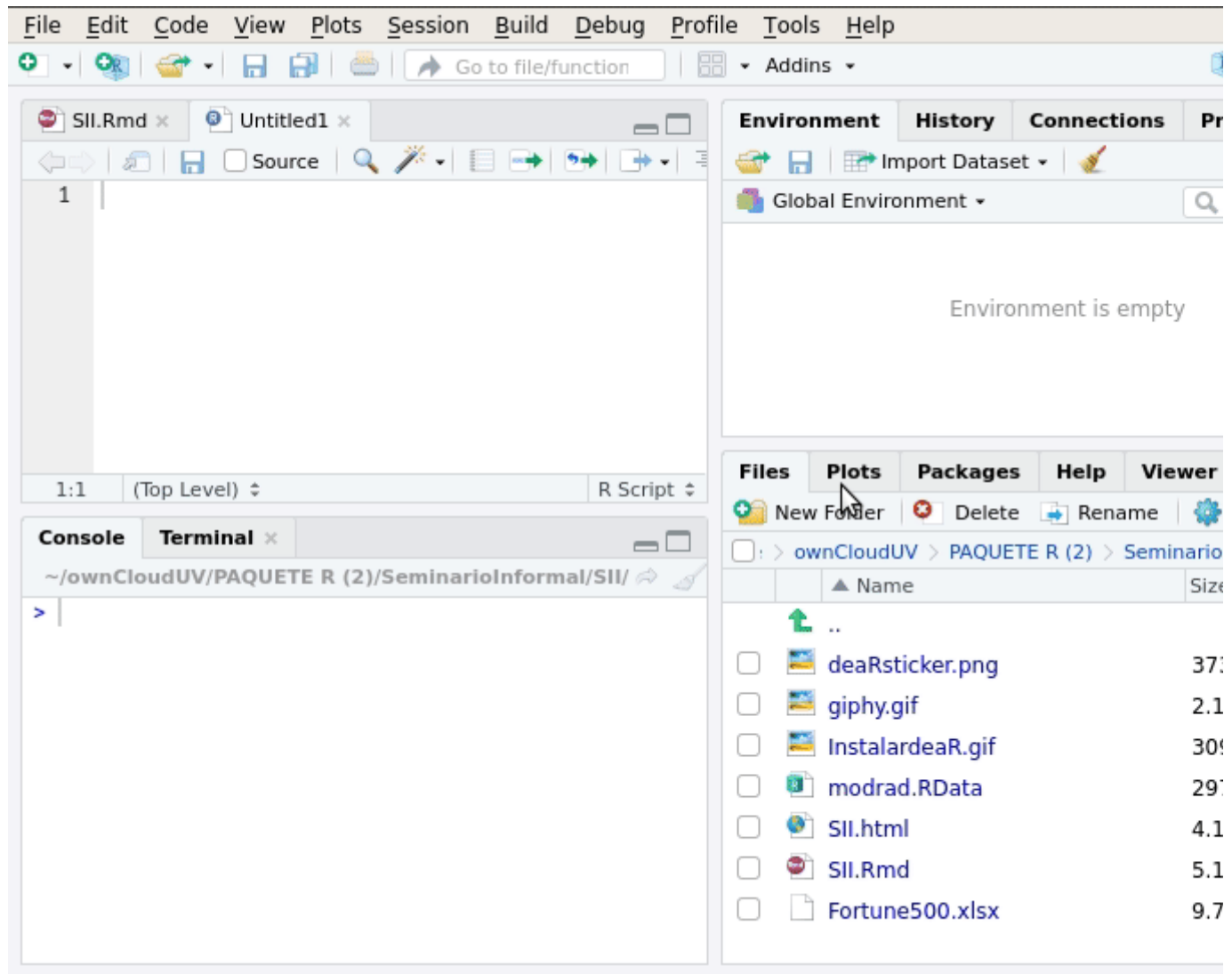


Workflow

Flujo de trabajo



Importar desde Excel



Leer los datos

- La tabla de datos hay que convertirlo en una lista **deadata**
- Usaremos la función **read_data**
- Hay que determinar las DMUs, los inputs / outputs y otras opciones

Company	Assets	Equity	Employees	Revenue	Profit
Mitsubishi	91920.6	10950.0	36000	184365.2	346.2
Mitsui	68770.9	5553.9	80000	181518.7	314.8
Itochu	65708.9	4271.1	7182	169164.6	121.2
General Motors	217123.4	23345.5	709000	168828.6	6880.7

Leer los datos

```
Fortune500 <- as.data.frame(Fortune500)
misdatos <- read_data(datadea = Fortune500, ni = 3, no = 2)
```

- **misdatos** es una lista de tipo **deadata** conteniendo

misdatos	list [9] (S3: deadata)	List of length 9
input	double [3 x 15]	91921 10950 36000 68771 5554 80000 65709 4271
output	double [2 x 15]	184365 346 181519 315 169165 121 168829 6881 1
dmunames	character [15]	'Mitsubishi' 'Mitsui' 'Itochu' 'General Motors' 'Sumiton
nc_inputs	NULL	Pairlist of length 0
nc_outputs	NULL	Pairlist of length 0
nd_inputs	NULL	Pairlist of length 0
nd_outputs	NULL	Pairlist of length 0
ud_inputs	NULL	Pairlist of length 0
ud_outputs	NULL	Pairlist of length 0

Aplicamos el modelo...

Modelo básico (input orientado)

$$\min_{\theta, \lambda} \quad \theta$$

s. a.:

$$\sum_{j=1}^n \lambda_j x_{ij} \leq \theta x_{io}, \quad i = 1, \dots, m$$

$$\sum_{j=1}^n \lambda_j y_{rj} \geq y_{ro}, \quad r = 1, \dots, s$$

$$\lambda_j \geq 0 \quad j = 1, \dots, n$$

Modelo básico con deaR

```
result <- model_basic(misdatos)
```

Extracción de resultados

- Parciales
 - efficiencies
 - lambdas
 - slacks
 - targets
 - multipliers
 - rts
 - references
- Generales
 - summary
 - plot

**Nos vamos al
ordenador...**

Otras opciones de
datos

I/O No controlables

- Ej. 2º input y 1er output no controlables

```
misdatos <- read_data(datadea = Fortune500, ni = 3, no = 2, nc
```

misdatos	list [9] (S3: deadata)	List of length 9
input	double [3 x 15]	91921 10950 36000 68771 5554 80000 65709 4271 7
output	double [2 x 15]	184365 346 181519 315 169165 121 168829 6881 16
dmunames	character [15]	'Mitsubishi' 'Mitsui' 'Itochu' 'General Motors' 'Sumitom
nc_inputs	double [1]	2
Equity	double [1]	2
nc_outputs	double [1]	1
Revenue	double [1]	1
nd_inputs	NULL	Pairlist of length 0
nd_outputs	NULL	Pairlist of length 0
ud_inputs	NULL	Pairlist of length 0
ud_outputs	NULL	Pairlist of length 0

Ejemplo (modelo básico)

$$\min_{\theta, \lambda} \quad \theta$$

s.a.:

$$\sum_{j=1}^n \lambda_j x_{ij} \leq \theta x_{io},$$

$$i \in IC \quad \sum_{j=1}^n \lambda_j x_{ij} = x_{io}, \quad i \notin IC$$

$$\sum_{j=1}^n \lambda_j y_{rj} \geq y_{ro},$$

$$r \in OC \quad \sum_{j=1}^n \lambda_j y_{rj} = y_{ro}, \quad r \notin OC$$

$$\lambda_j \geq 0$$

$$j = 1, \dots, n$$

```
result <- model_basic(misdatos)
```

I/O No discrecionales

- Ej. 1er y 3er inputs y 2º output no discrecionales

```
misdatos <- read_data(datadea = Fortune500, ni = 3, no = 2,  
                      nd_inputs = c(1,3), nd_outputs = 2)
```

misdatos	list [9] (S3: deadata)	List of length 9
input	double [3 x 15]	91921 10950 36000 68771 5554 80000 65709 4271
output	double [2 x 15]	184365 346 181519 315 169165 121 168829 6881 1
dmunames	character [15]	'Mitsubishi' 'Mitsui' 'Itochu' 'General Motors' 'Sumiton
nc_inputs	NULL	Pairlist of length 0
nc_outputs	NULL	Pairlist of length 0
nd_inputs	double [2]	1 3
Assets	double [1]	1
Employees	double [1]	3
nd_outputs	double [1]	2
Profit	double [1]	2
ud_inputs	NULL	Pairlist of length 0
ud_outputs	NULL	Pairlist of length 0

Ejemplo (modelo básico)

$$\min_{\theta, \lambda} \quad \theta$$

s. a.:

$$\sum_{j=1}^n \lambda_j x_{ij} \leq \theta x_{io},$$

$$\sum_{j=1}^n \lambda_j y_{rj} \geq y_{ro}$$

$$\lambda_j \geq 0$$

$$\sum_{\substack{j=1 \\ i \in ID}}^n \lambda_j x_{ij} \leq x_{io}, \quad i \notin ID$$

$$\sum_{\substack{j=1 \\ r \in OD}}^n \lambda_j y_{rj} \geq y_{ro}, \quad r \notin OD$$

$$j = 1, \dots, n$$

I/O No deseables (o bad)

- Se transforman los inputs/outputs no deseables (normalmente son outputs)
- Ejemplo: Huan-Bian, 2007. 2 Inputs, 3 outputs, output 3 no deseable

```
huan <- read_data(Hua_Bian_2007, ni = 2, no = 3, ud_outputs =  
results1 <- model_basic(huan, orientation = "oo", rts = "vrs")  
results2 <- model_basic(huan, orientation = "oo", rts = "vrs",
```

Otras opciones
generales de los
modelos

Otras opciones

- Orientación del modelo (**orientation**)
 - Input / Output / Direccional / No orientado
- Retornos a escala (**rts**)
 - CRS, VRS, NIRS, NDRS, GRS
- DMUs a evaluar (**dmu_eval**)
- DMUs de referencia (**dmu_ref**)
- 2ª etapa (**max_slack**) con pesos

Índice de Malmquist

Índice de Malmquist

- Tenemos datos en distintos periodos de tiempo
- Vemos la evolución de las eficiencias
- Los datos están formateados de manera diferente:
 - Tablas horizontales
 - Tablas verticales
- La función `read_malmquist` genera una lista de deadata

Tabla horizontal

DMUs	Capital2005	Labor2005	GIOV2005	Capital2006	Labor2006
Beijing	12829.79	116.97	6946.07	14244.40	117.36
Tianjin	6347.92	122.17	6774.10	7129.02	116.33
Hebei	9473.70	292.21	11008.12	11250.95	303.35
Shanxi_1	7045.09	213.20	4850.91	8865.50	220.59
Neimenggu	4595.89	83.70	2995.59	5605.92	90.72
Liaoning	11902.12	276.55	10814.51	14140.89	303.02

Tabla vertical

	DMUs	Period	Capital	Labor	GIOV
1	Beijing	2005	12829.79	116.97	6946.07
2	Tianjin	2005	6347.92	122.17	6774.10
...
31	Sinkiang	2005	2837.00	46.70	2102.53
32	Beijing	2006	14244.40	117.36	8210.00
33	Tianjin	2006	7129.02	116.33	8527.70
...

Lectura de datos (Tabla horizontal)

- Hace falta indicar el número de periodos, el número de inputs/outputs

```
ejemplo_malmquist <- read_malmquist(datadea = Economy,  
                                     nper = 5,  
                                     arrangement = "horizontal",  
                                     ni = 2,  
                                     no = 1)
```

ejemplo_malmquist	list [5]	List of length 5
Period.1	list [9] (S3: deadata)	List of length 9
input	double [2 x 31]	12829.79 116.97 6347.92 122.17 9473.70 292.21 7
output	double [1 x 31]	6946.1 6774.1 11008.1 4850.9 2995.6 10814.5 379
dmunames	factor	Factor with 31 levels: "Beijing", "Tianjin", "Hebei", "
nc_inputs	NULL	Pairlist of length 0
nc_outputs	NULL	Pairlist of length 0
nd_inputs	NULL	Pairlist of length 0
nd_outputs	NULL	Pairlist of length 0
ud_inputs	NULL	Pairlist of length 0
ud_outputs	NULL	Pairlist of length 0
Period.2	list [9] (S3: deadata)	List of length 9
Period.3	list [9] (S3: deadata)	List of length 9
Period.4	list [9] (S3: deadata)	List of length 9

Lectura de datos (Tabla vertical)

- Hace falta indicar la columna que contiene la información temporal, los inputs/outputs

```
ejemplo_malmquist2 <- read_malmquist(datadea = EconomyLong,  
                                     percol = 2,  
                                     arrangement = "vertical",  
                                     ni = 2,  
                                     no = 1)
```

ejemplo_malmquist2	list [5]	List of length 5
2005	list [9] (S3: deadata)	List of length 9
input	double [2 x 31]	2005 12830 2005 6348 2005 9474 2005 7045 200
output	double [1 x 31]	116.97 122.17 292.21 213.20 83.70 276.55 101.83
dmunames	factor	Factor with 31 levels: "Beijing", "Tianjin", "Hebei", '
nc_inputs	NULL	Pairlist of length 0
nc_outputs	NULL	Pairlist of length 0
nd_inputs	NULL	Pairlist of length 0
nd_outputs	NULL	Pairlist of length 0
ud_inputs	NULL	Pairlist of length 0
ud_outputs	NULL	Pairlist of length 0
2006	list [9] (S3: deadata)	List of length 9
2007	list [9] (S3: deadata)	List of length 9
2008	list [9] (S3: deadata)	List of length 9

Ejecutamos el modelo

```
MI <- malmquist_index(ejemplo_malmquist2)
resumen <- summary(MI)
knitr::kable(resumen$Results[1:5,])
```

Period	DMU	ec	tc	pech	sech	mi
2006	Beijing	0.9904260	0.9185664	1	0.9904260	0.9097720
2006	Tianjin	0.9049332	0.9383294	1	0.9049332	0.8491254
2006	Hebei	0.9502866	0.9329661	1	0.9502866	0.8865852
2006	Shanxi_1	0.8854319	0.9407446	1	0.8854319	0.8329653
2006	Neimenggu	0.9483924	0.9369410	1	0.9483924	0.8885877

Más resultados

```
knitr::kable(resumen$means_by_period)
```

Period	ec	tc	pech	sech	mi
2006	0.9500551	0.9350165	1	0.9500551	0.8883172
2007	0.9627391	0.9052274	1	0.9627391	0.8714978
2008	0.9440146	0.9441444	1	0.9440146	0.8912862
2009	1.0063403	0.8777664	1	1.0063403	0.8833317

Y más todavía...

```
knitr::kable(head(resumen$means_by_dmu))
```

DMU	ec	tc	pech	sech	mi
Anhui	0.9809305	0.9054903	1	0.9809305	0.8882230
Beijing	0.9925625	0.9174540	1	0.9925625	0.9106305
Chongqing	1.0149624	0.9054903	1	1.0149624	0.9190386
Fujian	0.9991211	0.9114593	1	0.9991211	0.9106582
Gansu	0.9154485	0.9054903	1	0.9154485	0.8289297
Guangdong	1.0000000	0.9939756	1	1.0000000	0.9939756

Eficiencia cruzada

Eficiencia cruzada

```
data_example <- read_data(datadea = Golany_Roll_1989,  
                           dmu = 1,  
                           inputs = 2:4,  
                           outputs = 5:6)  
result <- cross_efficiency(data_example,  
                           orientation = "io",  
                           rts = "crs",  
                           selfapp = TRUE)
```

Doyle, J.; Green, R. (1994). *"Efficiency and cross efficiency in DEA: derivations, meanings and the uses"*, Journal of Operational Research Society, 45(5), 567–578. DOI: 10.2307/2584392

Eficiencia cruzada (resultados)

Método arbitrario

	DMU_1	DMU_2	DMU_3	DMU_4
DMU_1	0.6810811	0.7179487	0.5973333	0.8129032
DMU_2	0.5518519	0.8333333	0.6196078	0.9000000
DMU_3	0.6666667	0.8333333	0.6274510	0.8571429
DMU_4	0.5444444	0.8312500	0.6176471	0.9000000

Eficiencia cruzada (resultados)

Método II (Agresivo)

	DMU_1	DMU_2	DMU_3	DMU_4
DMU_1	0.6810811	0.7179487	0.5973333	0.8129032
DMU_2	0.5518519	0.8333333	0.6196078	0.9000000
DMU_3	0.6666667	0.8333333	0.6274510	0.8571429
DMU_4	0.5444444	0.8312500	0.6176471	0.9000000

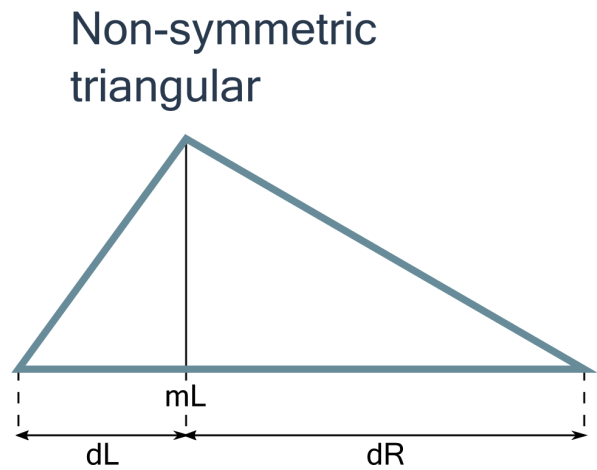
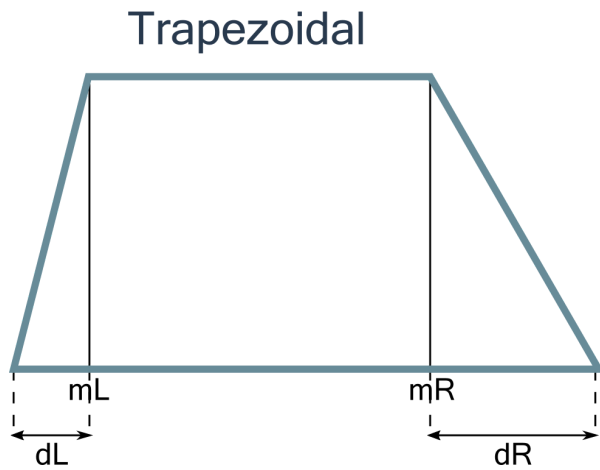
Eficiencia cruzada (resultados)

Método II (Benevolente)

	DMU_1	DMU_2	DMU_3	DMU_4
DMU_1	0.6810811	0.7179487	0.5973333	0.8129032
DMU_2	0.5518519	0.8333333	0.6196078	0.9000000
DMU_3	0.6666667	0.8333333	0.6274510	0.8571429
DMU_4	0.5444444	0.8312500	0.6176471	0.9000000

Modelos Fuzzy

Números fuzzy



Lectura de datos (Leon2003)

DMU	x	alpha	y	beta
A	3.0	2.0	3.0	1.00
B	4.0	0.5	2.5	1.00
C	4.5	1.5	6.0	1.00
D	6.5	0.5	4.0	1.25
E	7.0	2.0	5.0	0.50
F	8.0	0.5	3.5	0.50

Lectura de datos (Triangular simétrico)

```
data_exampleL <- read_data_fuzzy(datadea = Leon2003, dmus = 1,  
                                outputs.mL = 4, outputs.dL = 5)
```


Lectura de datos (Kao_Liu_2003)

Library	Patronage	Collections	Personel	Expenditures	Build
Library_01	0.28	0.32	0.20	0.41	0
Library_02	0.16	0.57	0.22	0.50	1
Library_03	1.00	1.00	1.00	1.00	0
Library_04	0.53	0.67	0.81	0.74	0
Library_05	0.02	0.10	0.08	0.15	0
Library_06	0.56	0.58	0.50	0.63	0
Library_07	0.24	0.41	0.34	0.58	0

Lectura de datos (Triangular asimétrico)

```
data_exampleKL <- read_data_fuzzy(datadea = Kao_Liu_2003, dmus
                                inputs.mL = 2,
                                outputs.mL = 3:7,
                                outputs.dL = c(NA, NA, 8, NA,
                                outputs.dR = c(NA, NA, 9, NA,
```

Corremos el modelo (Kao-Liu)

- Se puede ejecutar sobre cualquier otro modelo crisp
- Considera el mejor y el peor caso
- Devuelve eficiencias para ciertos α -cortes

```
result <- modelfuzzy_kaoliu(data_exampleL,  
                             kaoliu_modelname = "basic",  
                             orientation = "io",  
                             rts = "crs",  
                             alpha = c(0,0.25,0.5,0.75,1))
```

Resultados (Peor caso)

	0	0.25	0.5	0.75	1
A	0.171	0.250	0.361	0.519	0.750
B	0.083	0.160	0.269	0.360	0.469
C	0.208	0.373	0.599	0.907	1.000
D	0.098	0.178	0.286	0.367	0.462
E	0.125	0.218	0.339	0.429	0.536
F	0.088	0.149	0.225	0.274	0.328

Resultados (Mejor caso)

	0	0.25	0.5	0.75	1
A	1.00	1.000	1.000	1.000	0.750
B	1.00	0.961	0.764	0.602	0.469
C	1.00	1.000	1.000	1.000	1.000
D	1.00	0.864	0.706	0.574	0.462
E	1.00	1.000	0.835	0.668	0.536
F	0.64	0.544	0.462	0.390	0.328

Próximamente...
Shiny!!!

Muchas gracias!!!