Package 'MultiplierDEA'

April 27, 2018

Type Package

Title Multiplier Data Envelopment Analysis and Cross Efficiency

Version 0.1.18

Date 2018-04-27

Author Aurobindh Kalathil Puthanpura <kalat2@pdx.edu>

Maintainer Aurobindh Kalathil Puthanpura <kalat2@pdx.edu>

Description
Functions are provided for calculating efficiency using multiplier DEA (Data Envelopment Analysis): Measuring the efficiency of decision making units (Charnes et al., 1978 <doi:10.1016/0377-2217(78)90138-8>) and cross efficiency using single and two-phase approach. In addition, it includes some datasets for calculating efficiency and cross efficiency.

License LGPL-2

Repository CRAN

Depends lpSolveAPI

NeedsCompilation no

LazyData true

Date/Publication 2018-04-27 17:24:13 UTC

R topics documented:

Bank_Branch_Operating_Efficiency

Data: Bank Branch Operating Efficiency data

Description

Bank Branch data for Operating Efficiency.

Usage

```
Bank_Branch_Operating_Efficiency
```

Format

A data frame containing data for 17 Bank Branches.

Branch_Code a character vector

PH a numeric vector

OE a numeric vector

SQM a numeric vector

A a numeric vector

B a numeric vector

C a numeric vector

D a numeric vector

Source

Giokas DI (1991) Bank branck operating efficiency: A comparative application of DEA and the Loglinear model, OMEGA International Journal of Management Science, 19 (6) 549-557.

References

Giokas DI (1991) Bank branck operating efficiency: A comparative application of DEA and the Loglinear model, OMEGA International Journal of Management Science, 19 (6) 549-557.

Examples

```
data(Bank_Branch_Operating_Efficiency)
attach(Bank_Branch_Operating_Efficiency)
x <- data.frame(PH, OE, SQM)
rownames(x) <- Branch_Code
colnames(x) <- colnames(Bank_Branch_Operating_Efficiency)[2:4]
y <- data.frame(A, B, C, D)
rownames(y) <- Branch_Code
colnames(y) <- colnames(Bank_Branch_Operating_Efficiency)[5:8]
```

```
detach(Bank_Branch_Operating_Efficiency)
# For CRS
result_CRS <- DeaMultiplierModel(x,y,"crs", "input")
# For VRS
result_VRS <- DeaMultiplierModel(x,y,"crs", "input")
```

BenchMark_Tests_And_Microcomputer
Data: Relationship between benchmark tests and Microcomputer price data

---

Description

The Relationship between benchmark tests and Microcomputer price data.

Usage

```
BenchMark_Tests_And_Microcomputer
```

Format

A data frame containing data for 22 Microcomputers.

System a character vector

Price a numeric vector

MemorySize a numeric vector

DiskCapacity a numeric vector

CPU a numeric vector

IO a numeric vector

RL1 a numeric vector

RL2 a numeric vector

RL3 a numeric vector

Source

Sircar S. and Dave D (1986) Tbe relationship between benchmark tests and microcomputer price. Communications of the ACM, 29, 212-217.

References

Sircar S. and Dave D (1986) Tbe relationship between benchmark tests and microcomputer price. Communications of the ACM, 29, 212-217.

Examples

```
data(BenchMark_Tests_And_Microcomputer)
attach(BenchMark_Tests_And_Microcomputer)

x <- BenchMark_Tests_And_Microcomputer

detach(BenchMark_Tests_And_Microcomputer)
```

CrossEfficiency   *Cross Efficiency Model*

Description

Cross Efficiency uses DEA to do peer evaluation of DMUs. Single-phase cross efficiency approach.

Usage

```
CrossEfficiency(x = x, y = y, rts = "crs", orientation = "input", weightRestriction)
```

Arguments

x	Inputs or resources used by each decision making unit.
y	Outputs or resources used by each decision making unit.
rts	Returns to scale for the application, or industry studied. Note the default rts is crs. vrs Variable returns to scale. crs Constant returns to scale.
orientation	Orientation of the DEA model - primary emphasis on input-reduction input or output-augmentation output. Note that unlike the DEA functions, the default is input orientation.
weightRestriction	
	Weight restriction for the model. Optional parameter.

Value

The function returns a number of values per DMU.

$ceva_matrix	Returns the cross efficiency matrix. Row is the Rating DMU and Column is the Rated DMU.
$ce_ave	Returns the cross efficiency score for the DMU.
$ceva_max	Returns the maximum cross efficiency score for the DMU.
$ceva_min	Returns the minimum cross efficiency score for the DMU.
$vx	Input weights from the model.
$uy	Output weights from the model.
$Model_Status	Returns the status of the LP model.

Note

ceva_matrix - cross-evaluation matrix. ceva_max - cross-evaluation maximum. ceva_min - cross-evaluation minimum. ce_ave - cross-efficiency scores.

Examples

```
#Example from Kenneth R. Baker: Optimization Modeling with Spreadsheets, Third Edition,p. 176,
#John Wiley and Sons, Inc.

dmu <- c("A", "B", "C", "D", "E", "F")
x <- data.frame(c(150,400,320,520,350,320),c(0.2,0.7,1.2,2.0,1.2,0.7))
rownames(x) <- dmu
colnames(x)[1] <- c("StartHours")
colnames(x)[2] <- c("Supplies")

y <- data.frame(c(14,14,42,28,19,14),c(3.5,21,10.5,42,25,15))
rownames(y) <- dmu
colnames(y)[1] <- c("Reimbursed")
colnames(y)[2] <- c("Private")

# Calculate the efficiency score
result <- CrossEfficiency(x,y,"crs", "input")
# Examine the cross efficiency score for DMUs
print(result$ce_ave)
```

DeaMultiplierModel *DEA Multiplier Model*

Description

DEA multiplier model calculates the efficieny and reference sets for each DMUs.

Usage

```
DeaMultiplierModel(x = x, y = y, rts = "crs", orientation = "input", weightRestriction)
```

Arguments

x	Inputs or resources used by each decision making unit.
y	Outputs or resources used by each decision making unit.
rts	Returns to scale for the application, or industry studied. Note the default rts is crs. vrs Variable returns to scale. crs Constant returns to scale. Available option: crs, vrs
orientation	Orientation of the DEA model - primary emphasis on input-reduction or output-augmentation output. Note that unlike the DEA functions, the default is input orientation. Available option: input, output.
weightRestriction	
	Weight restriction for the model. Optional parameter.

Value

The function returns a number of values per DMU. The standardized efficiency (all inefficiencies are between 0 and 1, for input and output orientation). Efficiency, and lambda values are returned.

$rts	Returns to scale of the model.
$Orientation	Orientation of the model.
$InputValues	Input Values (x) passed to the model.
$OutputValues	Output Values (y) passed to the model.
$Efficiency	Efficiency of each DMU in the model.
$Lambda	Lambdas per DMU in the model.
$HCU_Input	HCU data for inputs.
$HCU_Output	HCU data for outputs.
$vx	Input weights from the model.
$uy	Output weights from the model.
$Free_Weights	Free weights from the model. Applies only to vrs returns-to-scale.
$Model_Status	Returns the status of the LP model.

Examples

```
#Example from Kenneth R. Baker: Optimization Modeling with Spreadsheets, Third Edition,p. 176,
#John Wiley and Sons, Inc.

dmu <- c("A", "B", "C", "D", "E", "F")

x <- data.frame(c(150,400,320,520,350,320),c(0.2,0.7,1.2,2.0,1.2,0.7))
rownames(x) <- dmu
colnames(x)[1] <- c("StartHours")
colnames(x)[2] <- c("Supplies")

y <- data.frame(c(14,14,42,28,19,14),c(3.5,21,10.5,42,25,15))
rownames(y) <- dmu
colnames(y)[1] <- c("Reimbursed")
colnames(y)[2] <- c("Private")

#Creating the weight restriction data frame with Upper bound

weightRestriction<-data.frame(lower = c(1), numerator = c("StartHours"),
denominator = c("Supplies"), upper = c(2))

#Creating the weight restriction data frame without Upper bound
weightRestriction<-data.frame(lower = c(1), numerator = c("StartHours"),
denominator = c("Supplies"))

#Creating the weight restriction data frame with Upper bound and Na, Inf or NaN
weightRestriction<-data.frame(lower = c(1,2), numerator = c("StartHours","Reimbursed"),
denominator = c("Supplies","Private"), upper = c(2,Inf))
```

```
# Calculate the efficiency score without weight Restriction
result <- DeaMultiplierModel(x,y,"crs", "input")
# Examine the efficiency score for DMUs
print(result$Efficiency)


# Calculate the efficiency score with weight Restriction
result <- DeaMultiplierModel(x,y,"crs", "input", weightRestriction)
# Examine the efficiency score for DMUs
print(result$Efficiency)
```

Departments_Of_Accounting

Data: UK University Departments Of Accounting Efficiency data.

Description

Evaluation the Efficiency of UK University Departments Of Accounting Efficiency.

Usage

```
Departments_Of_Accounting
```

Format

A data frame containing data for 20 UK University Departments Of Accounting.

Departments a numeric vector

Undergraduates a numeric vector

Research a numeric vector

Taught a numeric vector

Res.Co a numeric vector

OtherRes a numeric vector

OtherIncome a numeric vector

Publications a numeric vector

AcademicStaff a numeric vector

Salaries a numeric vector

OtherExp a numeric vector

Source

Tomkins C and Green RH (1988) An experiment in the use of data envelopment analysis for evaluating the efficiency of UK university departments of accounting. Financial Accounting and Management, 4, 147-164.

References

Tomkins C and Green RH (1988) An experiment in the use of data envelopment analysis for evaluating the efficiency of UK university departments of accounting. Financial Accounting and Management, 4, 147-164.

Examples

```
data(Departments_Of_Accounting)
attach(Departments_Of_Accounting)

x <- data.frame(AcademicStaff)
rownames(x) <- Departments
colnames(x) <- colnames(Departments_Of_Accounting)[9]

y <- data.frame(Undergraduates, Research, Taught,(Res.Co + OtherRes + OtherIncome))
rownames(y) <- Departments
colnames(y)[1] <- colnames(Departments_Of_Accounting)[2]
colnames(y)[2] <- colnames(Departments_Of_Accounting)[3]
colnames(y)[3] <- colnames(Departments_Of_Accounting)[4]
colnames(y)[4] <- c("Total_Income")

detach(Departments_Of_Accounting)

result <- DeaMultiplierModel(x,y,"crs", "input")
```

dict.solveStatus	_Provides the solver status codes._

Description

Provides the solver status codes and description.

Examples

```
#List status codes and description.

dict.solveStatus
```

Evaluations_Of_NonProfitOrganizations	
	Data: Evaluation of Non-Profit organizations data

Description

Evaluation of Non-Profit organizations efficiency.

Usage

```
Evaluations_Of_NonProfitOrganizations
```

Format

A data frame containing data for 16 Non-Profit organizations.

`Hospital` a numeric vector

`H0` a numeric vector

`PercentOccupancy` a numeric vector

`RevenuePerDay` a numeric vector

`A/RTurnover` a numeric vector

`CostPerDay` a numeric vector

`LengthOfStay` a numeric vector

Source

Greenberg R and Nunamaker T (1987) A generalized multiple criteria model for control and evaluation of nonprofit organizations. Financial Accountability and Management, 3 (4), 331-342.

References

Greenberg R and Nunamaker T (1987) A generalized multiple criteria model for control and evaluation of nonprofit organizations. Financial Accountability and Management, 3 (4), 331-342.

Examples

```
data(Evaluations_Of_NonProfitOrganizations)
attach(Evaluations_Of_NonProfitOrganizations)

x <- Evaluations_Of_NonProfitOrganizations

detach(Evaluations_Of_NonProfitOrganizations)
```

Evaluation_Educational_Program

Data: Educational program data

Description

Evaluation of Educational program.

Usage

```
Evaluation_Educational_Program
```

Format

A data frame containing data for 22 educational programs.

Program a numeric vector

CCR_EFF a numeric vector

Revenue_Generated a numeric vector

Student_Employed a numeric vector

Employer_Satisfaction a numeric vector

Contact_Hours a numeric vector

Number_of_FTE_Staff a numeric vector

Facility_Allocation a numeric vector

Expenditures a numeric vector

Source

Bessent A, Bessent W, Cbames A, Cooper WW and Thorgood N (1983) Evaluation of educational program proposals by means of data envelopment analysis. Educational Administrative Quarterly, 19, 82-107.

References

Bessent A, Bessent W, Cbames A, Cooper WW and Thorgood N (1983) Evaluation of educational program proposals by means of data envelopment analysis. Educational Administrative Quarterly, 19, 82-107.

Examples

```
data(Evaluation_Educational_Program)
attach(Evaluation_Educational_Program)


x <- data.frame(Contact_Hours, Number_of_FTE_Staff, Facility_Allocation, Expenditures)
rownames(x) <- Program
colnames(x) <- colnames(Evaluation_Educational_Program)[6:9]


y <- data.frame(Revenue_Generated, Student_Employed, Employer_Satisfaction)
rownames(y) <- Program
colnames(y) <- colnames(Evaluation_Educational_Program)[3:5]

detach(Evaluation_Educational_Program)

result <- DeaMultiplierModel(x,y,"crs", "input")
```

Mal_Ben   *Benevolent and Malevolent Model*

Description

Two-Phase Cross efficiency approach.

Usage

```
Mal_Ben(x = x, y = y, rts ="crs", orientation = "input", phase = "mal",
weightRestriction, include = TRUE)
```

Arguments

x	Inputs or resources used by each decision making unit.
y	Outputs or resources used by each decision making unit.
rts	Returns to scale for the application, or industry studied. Note the default rts is crs. vrs Variable returns to scale. crs Constant returns to scale. Available option: crs, vrs.
orientation	Orientation of the DEA model - primary emphasis on input-reduction input or output-augmentation output. Note that unlike the DEA functions, the default is input orientation. Available option: input, output.
weightRestriction	
	Weight restriction for the model. Optional parameter.
phase	Second phase of the model. Malevolent or Benevolent. Note the default is mal.Available option: mal, ben.
include	In the second phase include evaluating DMU in the calculation. Default is TRUE. Available option: TRUE, FALSE.

Value

The function returns a number of values per DMU. The standardized efficiency (all inefficiencies are between 0 and 1, for input and output orientation) Efficiency, and the lambda values, lambda, are returned.

$rts	Returns to scale of the model.
$Orientation	Orientation of the model.
$InputValues	Input Values (x) passed to the model.
$OutputValues	Output Values (y) passed to the model.
$Phase1_Efficiency	
	Efficiency of each DMU in the model from Phase 1.
$Phase1_Lambda	
	Lambdas per DMU in the model from Phase 1.
$Phase1_vx	Input weights from the model from Phase 1.

$Phase1_uy   Output weights from the model from Phase 1.

$Phase1_Free_Weights

                    Free weights from the model from Phase 1. Applies only to vrs returns-to-scale.

$Phase1_Model_Status

                    Returns the status of the phase two LP model.

$Phase2_Efficiency

                    Efficiency of each DMU in the model from Phase 2.

$Phase2_Lambda

                    Lambdas per DMU in the model from Phase 2.

$Phase2_vx   Input weights from the model from Phase 2.

$Phase2_uy   Output weights from the model from Phase 2.

$Phase2_Free_weights

                    Free weights from the model from Phase 2. Applies only to vrs returns-to-scale.

$Phase2_Model_Status

                    Returns the status of the phase two LP model.

$ceva_matrix   Returns the cross efficiency matrix. Row is the Rating DMU and Column is the
                    Rated DMU.

$ce_ave   Returns the cross efficiency score for the DMU.

$ceva_max   Returns the maximum cross efficiency score for the DMU.

$ceva_min   Returns the minimum cross efficiency score for the DMU.

Note

ceva_matrix - cross-evaluation matrix. ceva_max - cross-evaluation maximum. ceva_min - cross-evaluation minimum. ce_ave - cross-efficiency scores.

Examples

```
#Example from Kenneth R. Baker: Optimization Modeling with Spreadsheets, Third Edition,p. 176,
#John Wiley and Sons, Inc.

dmu <- c("A", "B", "C", "D", "E", "F")
x <- data.frame(c(150,400,320,520,350,320),c(0.2,0.7,1.2,2.0,1.2,0.7))
rownames(x) <- dmu
colnames(x)[1] <- c("StartHours")
colnames(x)[2] <- c("Supplies")
y <- data.frame(c(14,14,42,28,19,14),c(3.5,21,10.5,42,25,15))
rownames(y) <- dmu
colnames(y)[1] <- c("Reimbursed")
colnames(y)[2] <- c("Private")

# Calculate the efficiency score
result <- Mal_Ben(x, y, rts = "crs",  orientation = "input",phase = "mal", include = TRUE)
# Examine the cross efficiency score for DMUs
print(result$ce_ave)
```

Metropolitan_And_London_Rates_Departments
Data: Metropolitan and London rates departments data

Description

Relative Efficiency Metropolitan and London rates departments.

Usage

```
Metropolitan_And_London_Rates_Departments
```

Format

A data frame containing data for 62 rates department authority.

Authority a character vector

TotalCost a numeric vector

Non-cnl a numeric vector

Rate a numeric vector

Summons a numeric vector

NPV a numeric vector

Source

Dyson RG and Thanassoulis E (1988) Reducing weight flexibility in Data Envelopment Analysis, Journal of the Operational Research Society, 39 (6), 563-576.

References

Dyson RG and Thanassoulis E (1988) Reducing weight flexibility in Data Envelopment Analysis, Journal of the Operational Research Society, 39 (6), 563-576.

Examples

```
data(Metropolitan_And_London_Rates_Departments)
attach(Metropolitan_And_London_Rates_Departments)

x <- data.frame(TotalCost)
rownames(x) <- Authority
colnames(x) <- colnames(Metropolitan_And_London_Rates_Departments)[2]


y <- data.frame(`Non-cnl`, Rate, Summons, NPV)
rownames(y) <- Authority
colnames(y) <- colnames(Metropolitan_And_London_Rates_Departments)[3:6]
```

```
detach(Metropolitan_And_London_Rates_Departments)

result <- DeaMultiplierModel(x,y,"crs", "input")
```

---

options.orientation.l *Provides the orientation option.*

Description

Provides the orientation option values.

Examples

```
# List the orientation option used as arguments.

options.orientation.l
```

---

options.phase.l   *Provides the second phase options.*

Description

Provides the second phase options available for Mal_Ben function.

Examples

```
# List the phase option used as arguments.

options.phase.l
```

---

options.rts.l   *Provides the rts (returns to scale) option.*

Description

Provides the rts (returns to scale) option values.

Examples

```
# List the returns to scale option used as arguments.

options.rts.l
```

Index

15