Package ‘frontiles’

April 2, 2023

Type Package
Title Partial Frontier Efficiency Analysis
Encoding latin1
Version 1.3
Date 2023-03-31
Depends R (>= 4.0.0), methods
Imports colorspace, rgl
License GPL (>= 2)
LazyLoad yes
Collate alphafrontier.2d.r alphafrontier.3d.r alphascore.r
ordermscore.r ordermscore.boot.r ordermfrontier.2d.r ROCscore.r
NeedsCompilation yes
Author Thibault Laurent [aut, cre, cph],
AbdelGaati Daouia [aut]
Maintainer Thibault Laurent <thibault.laurent@univ-tlse1.fr>
Repository CRAN
Date/Publication 2023-04-02 15:20:02 UTC

R topics documented:

frontiles-package .......................................................... 2
alphafrontier.2d ............................................................. 3
alphafrontier.3d ............................................................. 4
alphascore ................................................................. 6
burposte ................................................................. 7
ordermfrontier.2d .......................................................... 8
Description

It calculates the alpha-quantile efficiency score and order-m score in multi-dimension and gives representation of alpha-quantile efficiency frontier. This work was supported by the agence nationale de la recherche through the EPI project (ANR-08-BLAN-0106-01)

Details

Package: frontiles
Type: Package
Version: 1.2
Date: 2023-04-01
License: GPL Version 2 or later
LazyLoad: yes

Author(s)

Abdelaati Daouia and Thibault Laurent
Maintainer: <thibault.laurent@univ-tlse1.fr>

References


Examples

data(spain)
xobs=as.matrix(spain[,3:4])
yobs=as.matrix(spain[,1])
score.ref.1<-alphascore(xobs,yobs)
Description

Representation of the alpha-quantile efficiency frontier (output, input or hyperbolic direction) for a set of reference points (xobs, yobs) in 2D (1 output and 1 input).

Usage

alphafrontier.2d(xobs, yobs, type="output", alpha=0.95, add=FALSE, confidence=FALSE, shade=FALSE,...)

Arguments

- xobs: a matrix of size \( n_1 \times 1 \), input of sample points
- yobs: a matrix of size \( n_1 \times 1 \), output of sample points
- type: a direction to choose among "output", "input" and "hyper"
- alpha: a scalar between 0 and 1
- add: a boolean with TRUE for keeping the active device
- confidence: a boolean for representing a confidence interval
- shade: a boolean for shading the confidence interval
- ...: usual options for plotting the frontier, lty, col, etc.

Details

Actually, there is no confidence interval when type="hyper". If type="input" and confidence=TRUE, the y-axis is permuted with the x-axis

Value

No return value, used for plotting the alpha-quantile efficiency frontier for 1 output and 1 input

Author(s)

Abdelaati Daouia and Thibault Laurent

References


See Also

ordermfrontier.2d, alphascore
Examples

# 1st example
data(spain)
plot(y ~ x2, data = spain)
alphafrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "output",
    alpha = 0.95, col = 'red', lty = 2, add = TRUE)
alphafrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "input",
    alpha = 0.95, col = 'royalblue', lty = 3, add = TRUE)
alphafrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "hyper",
    alpha = 0.95, col = 'green', lty = 4, add = TRUE)
legend("topleft", title = "alpha-quantile frontier; alpha=0.95",
    legend = c("output direction", "input direction", "hyper direction"),
    lty = 2:4, col = c("red", "royalblue", "green"))

# 2nd example
plot(y ~ x2, data = spain)
alphafrontier.2d(as.matrix(spain$x2), as.matrix(spain$y),
    type = "output", alpha = 1, add = TRUE)
alphafrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "output",
    alpha = 0.95, col = 'blue', lty = 2, add = TRUE)
ordermfrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "output",
    m = 30, col = 'green', lty = 3, add = TRUE)
legend("topleft", title = "output direction", legend = c("FDH", "alpha=0.95", "m=30"),
    lty = 1:3, col = c("black", "royalblue", "green"))

# 3rd example
alphafrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "output",
    confidence = TRUE, shade = TRUE, alpha = 0.98)
title("Alpha-quantile frontier with alpha=0.98 and its confidence interval")

---

alphafrontier.3d

3d representation of alpha-quantile frontier in the case of 2 input and 1 output.

Description

3d representation of alpha-quantile frontier for a set of reference points (xobs,yobs) in the case of two input and one output. No representation yet for hyperbolic direction.

Usage

alphafrontier.3d(xobs, yobs, type="output", alpha=0.95, digits=4,
    box.leg=TRUE, palette=heat_hcl, rgl=FALSE, n.class=NULL, ...)

Arguments

xobs a matrix of size $n_1 \times 2$, input of reference points
yobs a matrix of size $n_1 \times 1$, output of reference points
alphafrontier.3d

- **type**: a character, "output" or "input" direction
- **alpha**: a scalar
- **digits**: a precision parameter to compute the alpha-frontier
- **box.leg**: representation of a legend-box on the plot with the values of frontier
- **palette**: function to use for colors in case where option col has not been called. See package colorspace for more informations.
- **rgl**: a boolean, for output direction, representation of the graphic in 3d if TRUE
- **n.class**: a numeric, for output direction, the number of class
- **...**: usual parameters of function plot. Use the parameter asp to modify the scale of window...

### Details

In "input" direction: You choose a value of output in the legend box, the efficiency-frontier of the input is represented with the corresponding color on the 2d graphic. In "output" direction: For calculate the alpha-quantile efficiency output frontier everywhere, we have constructed a grid of size \( n \times n \) by drawing vertical and horizontal lines which intersect each reference observations. Then, we have calculated for each cell \( C_k \) \( k = 1, \ldots, n^2 \) the prediction of the alpha-quantile output efficiency frontier which correspond to the \( \alpha^{th} \) elements of the suite \( \{y_j\}_{j=1,\ldots,n_k} \) where reference observations \( j, j = 1, \ldots, n_k \) verify

\[
x_1^j \leq \inf_{(x_1,x_2) \in C_k} x_1
\]

and \( x_2^j \leq \inf_{(x_1,x_2) \in C_k} x_2 \).

We propose to use an algorithm which filled up cells with colors depending on the values taken by the alpha-quantile output efficiency frontier. The algorithm attach the row and vary the column as the following figure can show it.

### Value

- **no values**

### Note

The algorithm used is certainly not optimized. For a data set of 61 observations, the function necessits 15.17s on an Optiplex GX745 2 duo 2.13GHz under Windows Vista and probably bugs beyond a certain number of observation

### Author(s)

Abdelaati Daouia and Thibault Laurent

### See Also

- [alphascore](#)
Examples

data(spain)
xyn <- cbind(spain[, 3:4], spain[, 1])
xtab <- as.matrix(xyn[, c(1,2)])
ytab <- matrix(xyn[, 3])

# representation in 2-d

op <- par(no.readonly = TRUE) # the whole list of settable par's.
alphafrontier.3d(xtab, ytab, type = "output", alpha = 0.6, xlab = "input 1",
    ylab = "input 2", main = "blabla")
points(xtab, pch = 16)
par(op)

alphascore

Calculates alpha-quantile efficiency score

Description

Calculates alpha-quantile efficiency score (output, input and hyperbolic direction) for a set of evaluation points (xeval, yeval) depending on reference points (xobs, yobs).

Usage

alphascore(xobs, yobs, xeval=xobs, yeval=yobs, alpha=0.95)

Arguments

- xobs: a matrix of size $n_1 \times p$, input of sample points
- yobs: a matrix of size $n_1 \times q$, output of sample points
- xeval: a matrix of size $n_2 \times p$, input of assessment points
- yeval: a matrix of size $n_2 \times q$, output of assessment points
- alpha: a scalar

Details

A score between 0 and 1 means that DMU is inefficient. If DMU greater than 1, DMU is super-efficient.

Value

a data.frame object with the alpha-quantile efficiency score in:

- output: output direction
- input: input direction
- hyper: hyperbolic direction
# 1st example
data(span)
res.alqf <- alphascore(xobs = as.matrix(span[, c(2, 3, 4)]),
                      yobs = as.matrix(span[, 1]), alpha = 0.8)

# 2nd example
data(burposte)
bur.samp <- burposte[which(burposte$xinput < 50000), ]
ind.samp <- sample(nrow(bur.samp), 500)
xeval <- as.matrix(bur.samp[ind.samp[1:100], 2])
yeval <- as.matrix(bur.samp[ind.samp[1:100], 3])
xobs <- as.matrix(bur.samp[ind.samp[101:500], 2])
yobs <- as.matrix(bur.samp[ind.samp[101:500], 3])

alphafrontier.2d(xobs, yobs, alpha = 0.95)
points(xeval, yeval, pch = 16, col = 'red')
text(xeval, yeval, as.character(1:100), adj = 2, cex = 0.8)
score.new.0.95 <- alphascore(xobs, yobs, xeval, yeval, alpha = 0.95)
Format

A data frame with 9521 observations on the following 3 variables.

- `ident` a numeric vector
- `xinput` a numeric vector
- `yprod` a numeric vector

References


Examples

```r
data(burposte)
```

ordermfrontier.2d

` Representation of m-order efficiency frontier for 1 output and 1 input`

Description

Representation of the m-order efficiency score (output, input or hyperbolic direction) for a set of reference points (xobs, yobs) in 2D (1 output and 1 input).

Usage

```r
ordermfrontier.2d(xobs, yobs, type = "output", m = 30, add = FALSE,
                   confidence = FALSE, shade = FALSE,...)
```

Arguments

- `xobs` a matrix of size $n_1 \times 1$, input of sample points
- `yobs` a matrix of size $n_1 \times 1$, output of sample points
- `type` a direction to choose among "output", "input" and "hyper"
- `m` an integer
- `add` a boolean with TRUE for keeping the active device
- `confidence` a boolean for representing a confidence interval
- `shade` a boolean for shading the confidence interval
- `...` usual options for ploting the frontier, lty, col, etc.

Details

Actually, there is no confidence interval when type="hyper". If type="input" and confidence=TRUE, the y-axis is permuted with the x-axis.
**Value**

No return value, used for plotting the m-order efficiency frontier for 1 output and 1 input

**Author(s)**

Abdelaati Daouia and Thibault Laurent

**References**


**Examples**

```r
# 1st example
data(spain)
plot(y ~ x2, data = spain)
ordermfrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "output",
                  m = 30, col = 'red', lty = 2, add = TRUE)
ordermfrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "input", m = 30,
                  col = 'royalblue', lty = 3, add = TRUE)
ordermfrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "hyper", m = 30,
                  col = 'green', lty = 4, add = TRUE)
legend("topleft", legend = c("output direction", "input direction", "hyper direction"),
        lty = 2:4, col = c("red", "royalblue", "green"))

# 2nd example
ordermfrontier.2d(as.matrix(spain$x2), as.matrix(spain$y), type = "output",
                  confidence = TRUE, shade = TRUE, m = 30)
```

**ordermscore**

*Calculates order-m efficiency score with asymptotic formula*

**Description**

Calculates order-m efficiency score (output, input and hyperbolic direction) for a set of assessment points (xeval, yeval) depending on sample points (xobs, yobs), using the formulas of Daouia and Gijbels (2011).

**Usage**

```r
ordermscore(xobs, yobs, xeval=xobs, yeval=yobs, m=30)
```
Arguments

- `xobs`: a matrix of size $n_1 \times p$, input of sample points
- `yobs`: a matrix of size $n_1 \times q$, output of sample points
- `xeval`: a matrix of size $n_2 \times p$, input of assessment points
- `yeval`: a matrix of size $n_2 \times q$, output of assessment points
- `m`: an integer

Details

A score between 0 and 1 means that DMU is inefficient. If DMU greater than 1, DMU is super-efficient. The asymptotic formula of the order-m score are given in Daouia and Gijbels (2011).

Value

- a data.frame object with the order-m efficiency score in:
  - `output`: output direction
  - `input`: input direction
  - `hyper`: hyperbolic direction

Author(s)

Abdelaati Daouia and Thibault Laurent

References


See Also

`alphascore`, `ordermfrontier.2d`

Examples

```r
# 1st example
data(spain)
score.orderm <- ordermscore(xobs = as.matrix(spain[, c(2, 3, 4)]),
                            yobs = as.matrix(spain[, 1]))

# 2nd example
data(burposte)
ind.samp <- sample(nrow(burposte), 500)
xobs <- as.matrix(burposte[ind.samp[1:100], 2])
yobs <- as.matrix(burposte[ind.samp[1:100], 3])
xeval <- as.matrix(burposte[ind.samp[101:500], 2])
yeval <- as.matrix(burposte[ind.samp[101:500], 3])
score.orderm.2 <- ordermscore(xobs, yobs, xeval, yeval)
```
ordermscore.boot

Calculates order-m efficiency score with bootstrap algorithm

Description
Calculates order-m efficiency score (output, input and hyperbolic direction) for a set of assessment points (xeval, yeval) depending on sample points (xobs, yobs), using the initial algorithm of Cazals et al. (2002).

Usage

ordermscore.boot(xobs, yobs, xeval=xobs, yeval=yobs, m=30, B=200, m.move=FALSE)

Arguments

- xobs: a matrix of size $n_1 \times p$, input of sample points
- yobs: a matrix of size $n_1 \times q$, output of sample points
- xeval: a matrix of size $n_2 \times p$, input of assessment points
- yeval: a matrix of size $n_2 \times q$, output of assessment points
- m: an integer, the number of selected firms
- B: an integer, the number of replication
- m.move: a boolean, to choose different values of m

Details
This function computes the algorithm initially proposed by Cazals et al. (2002). If m.move=TRUE, different values of m are given as suggested by Daouia et al (2009).

Value

A data.frame object with the average mean order-m efficiency score and standard deviation associated:

- output: output direction
- output: output direction
- input: input direction
- input: input direction
- hyper: hyperbolic direction
- hyper: hyperbolic direction

Author(s)

Abdelaati Daouia and Thibault Laurent
ROCscore

The ROC curve to help choosing alpha and m parameters

Description

Computes the percentage of firms super-efficient according to the parameter alpha for alpha-quantile score and m for m-order score in a given direction.

Usage

ROCscore(xobs, yobs, type="output")
**Arguments**

- `xobs`: A matrix of size $n_1 \times p$, input of sample points
- `yobs`: A matrix of size $n_1 \times q$, output of sample points
- `type`: A direction to choose among "output", "input" and "hyper"

**Details**

A firm is super-efficient if its score is greater than 1. By consulting this graph, we may choose the values of alpha and m which correspond to the desired degree of robustness, i.e. the percentage of high performers of the population we want to exclude in our more realistic benchmarking comparison (see p.78 of Daraio and Simar, 2010).

**Value**

A data.frame object with:

- `alpha`: Different values of alpha
- `f(alpha)`: The percentage of firms super-efficient
- `m`: Different values of m
- `f(m)`: The percentage of firms super-efficient

**Author(s)**

Abdelaati Daouia and Thibault Laurent

**References**


**Examples**

```r
# 1st example
data(spain)
res.roc <- ROCscore(xobs = as.matrix(spain[,c(2,3,4)]),
yobs = as.matrix(spain[,1]),
type = "output")
```
Description
Spain data

Usage
data(spain)

Format
A data frame with 61 observations on the following 4 variables.

- y a numeric vector of output
- x1 a numeric vector of input
- x2 a numeric vector of input
- x3 a numeric vector of input

Examples
data(spain)
Index

* datasets
  - burposte, 7
  - spain, 14

* multivariate
  - alphafrontier.2d, 3
  - alphafrontier.3d, 4
  - alphascore, 6
  - ordermfrontier.2d, 8
  - ordermscore, 9
  - ordermscore.boot, 11
  - ROCscore, 12

* robust
  - alphafrontier.2d, 3
  - alphafrontier.3d, 4
  - alphascore, 6
  - ordermfrontier.2d, 8
  - ROCscore, 12

alphafrontier.2d, 3, 7
alphafrontier.3d, 4
alphascore, 3, 5, 6, 10, 12

burposte, 7

frontiles (frontiles-package), 2
frontiles-package, 2

ordermfrontier.2d, 3, 8, 10
ordermscore, 7, 9, 12
ordermscore.boot, 11

ROCscore, 12

spain, 14