Package 'CPP'

May 11, 2018

Type Package

Title Composition of Probabilistic Preferences (CPP)

Version 0.1.0

Author Luiz O. Gavião, Annibal P. Sant'Anna, Gilson B.A. Lima and Pauli A.A. Garcia.

Maintainer Luiz O. Gavião <luiz.gaviao67@gmail.com>

Description CPP is a multiple criteria decision method to evaluate alternatives on complex decision making problems, by a probabilistic approach. The CPP was created and expanded by Sant'Anna, Annibal P. (2015) <doi:10.1007/978-3-319-11277-0>.

Depends R (>= 2.1.0), ineq, kappalab, mc2d

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-11 10:06:13 UTC

R topics documented:

`CPP-package` *A Package for the Composition of Probabilistic Preferences (CPP)*

Description

This package supports the decision making in multicriteria problems. The CPP method uses a probabilistic approach to emulate the uncertainty that is inevitably present in preference evaluations. This uncertainty is modelled by probability distributions around the exact measurements of alternatives in all criteria considered in the problem. This package includes several functions described in the references, with a focus on ranking, sorting and analysing the evolution of alternatives.

Author(s)

Luiz Octavio Gaviao (luiz.gaviao67@gmail.com);

Annibal Parracho Sant'Anna (annibal.parracho@gmail.com);

Gilson Brito Alves Lima (glima@id.uff.br); and

Pauli Adriano de Almada Garcia (pauliadriano@gmail.com).

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Gaviao, Luiz O. & Lima, Gilson B.A. (2017) Support decision to player selection: an application of the CPP in soccer, Novas Edicoes Academicas [in Portuguese].

`Agg.Sim` *Aggregation of expert's estimatives by similarity of values*

Description

This function computes the aggregated value of different expert's estimatives, using Beta PERT distributions to randomize the decision matrix.

Usage

```
Agg.Sim(x, min, max, s, w, b)
```

Arguments

`x`	Decision matrix of expert estimatives (rows) and criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
`min`	Vector of minimum values in each criterion scale. For common scales to all criteria, the vector must repeat the minimum value as many times as the number of criteria.
`max`	Vector of maximum values in each criterion scale. For common scales to all criteria, the vector must repeat the maximum value as many times as the number of criteria.
`s`	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis of the random variable.
`w`	Weights describing the expert experience in the subject matter.
`b`	Beta describes the balance between the expert weights and their opinions. Beta varies in the interval [0,1]. The higher the index, the higher the importance of weights.

Value

SM are the Similarity Matrices per criterion. CDC describes the Consensus Coefficient matrix. Agg.value gives the aggregated value of expert opinions per criterion.

Examples

```
## Expert's estimatives on four criteria
Exp.1 = c(4,7,6,8)
Exp.2 = c(4,3,6,5)
Exp.3 = c(3,8,2,9)
Exp.4 = c(6,8,9,7)
Exp.5 = c(5,9,2,4)
Exp.6 = c(7,6,5,5)
x = rbind(Exp.1,Exp.2,Exp.3,Exp.4,Exp.5) # Decision matrix
min = c(0,0,0,0) # Minimum scale values.
max = c(10,10,10,10) # Maximum scale values.
```

```
s = 4 # Shape
w = c(0.4,0.3,0.2,0.06,0.04) # Expert relevance.
b = 0.4
Agg.Sim(x,min,max,s,w,b)
```

AHP.Beta *Probabilistic AHP using Beta PERT distributions*

Description

This function computes criteria weights, using AHP and randomic pair-wise evaluations by Beta PERT distributions.

Usage

```
AHP.Beta(n, s, list)
```

Arguments

n	Random numbers created from Beta PERT distributions, using the parameters 'min', 'mean' and 'max' of each pair-wise criteria comparison elicited from the experts.
s	Shape of a Beta PERT distribution, as described in package "mc2d". There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data elicited from experts.
list	List of pair-wise comparison matrices of expert opinions. The function 'list' is embedded in R.

Value

Weights returned from a simulation of AHP with Beta PERT distributions. The weights are driven from the simulated matrix that gives the minimum AHP Consistent Ratio.

References

Saaty, Thomas L. (1980). The analytic hierarchy process: planning, priority setting, resource allocation, McGraw-Hill.

Examples

```
n=5000 # simulation
s=6 # shape of Beta PERT distribution
# Expert pair-wise evaluations
Exp.1 = matrix(c(1,0.2,0.3,5,1,0.2,3,5,1),3,3)
Exp.2 = matrix(c(1,2,8,0.5,1,6,0.12,0.16,1),3,3)
Exp.3 = matrix(c(1,0.5,0.5,2,1,6,2,0.16,1),3,3)
Exp.4 = matrix(c(1,3,4,0.3,1,0.5,0.25,0.3,1),3,3)
Exp.5 = matrix(c(1,4,5,0.25,1,1,0.2,1,1),3,3)
list = list(Exp.1,Exp.2,Exp.3,Exp.4,Exp.5)
AHP.Beta(n,s,list)
```

AHP.Unif *Probabilistic AHP using Uniform distributions*

Description

This function computes criteria weights, using AHP and randomic pair-wise evaluations by Uniform distributions.

Usage

```
AHP.Unif(n, list)
```

Arguments

n	Random numbers created from Uniform distributions, using the parameters 'min' and 'max' of each pair-wise criteria comparison elicited from the experts.
list	Pair-wise comparison matrices of expert opinions. The function 'list' is embedded in R.

Value

Weights returned from a simulation of AHP with Uniform distributions. The weights are driven from the simulated matrix that gives the minimum AHP Consistent Index.

References

Saaty, Thomas L. (1980). The analytic hierarchy process: planning, priority setting, resource allocation, McGraw-Hill.

Examples

```
n=5000 # Simulation
# Expert pair-wise evaluations
Exp.1 = matrix(c(1,0.2,0.3,5,1,0.2,3,5,1),3,3)
Exp.2 = matrix(c(1,2,8,0.5,1,6,0.12,0.16,1),3,3)
Exp.3 = matrix(c(1,0.5,0.5,2,1,6,2,0.16,1),3,3)
Exp.4 = matrix(c(1,3,4,0.3,1,0.5,0.25,0.3,1),3,3)
Exp.5 = matrix(c(1,4,5,0.25,1,1,0.2,1,1),3,3)
list = list(Exp.1,Exp.2,Exp.3,Exp.4,Exp.5)
AHP.Unif(n,list)
```

`CPP.AHP.Beta` *CPP Additive Weighting with Probabilistic AHP using Beta PERT distributions*

Description

This function computes CPP by additive weighting. Experts' estimatives are based on pair-wise comparisons of criteria and are joined in a list of matrices. The estimatives are used as parameters to probabilistic distributions. The minimum, mean, and maximum values of each pair of criteria are used to model Beta PERT distributions. Randomic values are generated and applied to the AHP method. The matrix that comprises de minimum AHP Consistent Index is used to return the criteria weights.

Usage

```
CPP.AHP.Beta(n, s, list, x)
```

Arguments

`n`	Random numbers created from Beta PERT distributions, using the parameters 'min', 'mean' and 'max' of each pair-wise criteria comparison elicited from the experts.
`s`	Shape of a Beta PERT distribution, as described in Package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data elicited from experts.
`list`	Pair-wise comparison matrices of expert opinions. The function 'list' is embedded in R.
`x`	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.

Value

Weights returned from the AHP method. PMax are the joint probabilities of each alternative being higher than the others, per criterion. CPP gives the final scores and ranks of alternatives by weighted sum.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Saaty, Thomas L. (1980). The analytic hierarchy process: planning, priority setting, resource allocation, McGraw-Hill.

Examples

```
n=5000 # simulation
s=6 # shape of Beta PERT distribution
# Expert pair-wise evaluations
Exp.1 = matrix(c(1,0.2,0.3,5,1,0.2,3,5,1),3,3)
Exp.2 = matrix(c(1,2,8,0.5,1,6,0.12,0.16,1),3,3)
Exp.3 = matrix(c(1,0.5,0.5,2,1,6,2,0.16,1),3,3)
Exp.4 = matrix(c(1,3,4,0.3,1,0.5,0.25,0.3,1),3,3)
Exp.5 = matrix(c(1,4,5,0.25,1,1,0.2,1,1),3,3)
list = list(Exp.1,Exp.2,Exp.3,Exp.4,Exp.5)
# Alternatives' original scores
Alt.1 = c(30,86,-5)
Alt.2 = c(26,77,-12)
Alt.3 = c(22,93,-4)
Alt.4 = c(34,65,-10)
Alt.5 = c(31,80,-8)
Alt.6 = c(29,79,-9)
Alt.7 = c(37,55,-15)
Alt.8 = c(21,69,-11)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8) # Decision matrix
CPP.AHP.Beta(n,s,list,x)
```

`CPP.AHP.Unif` *CPP Additive Weighting with Probabilistic AHP using Uniform distributions*

Description

This function computes CPP by additive weighting. Experts' estimatives are based on pair-wise comparisons of criteria and are joined in a list of matrices. The estimatives are used as parameters to probabilistic distributions. The minimum and maximum values of each pair of criteria are used to model Uniform distributions. Randomic values are generated and applied to the AHP method. The matrix that comprises de minimum AHP Consistent Index is used to return the criteria weights.

Usage

```
CPP.AHP.Unif(n, list, x)
```

Arguments

`n`	Random numbers based on Uniform distributions, using the parameters 'min' and 'max' of each pair-wise criteria comparison elicited from the experts.
`list`	Pair-wise comparison matrices of expert opinions. The function 'list' is embedded in R.
`x`	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.

Value

Weights returned from the AHP method. PMax are the joint probabilities of each alternative being higher than the others, per criterion. CPP gives the final scores and ranks of alternatives by weighted sum.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer

Saaty, Thomas L. (1980). The analytic hierarchy process: planning, priority setting, resource allocation, McGraw-Hill.

Examples

```
# Computing weights by the AHP method, with 'n' simulated matrices.
n=5000 # simulation
# Expert pair-wise evaluations
Exp.1 = matrix(c(1,0.2,0.3,5,1,0.2,3,5,1),3,3)
Exp.2 = matrix(c(1,2,8,0.5,1,6,0.12,0.16,1),3,3)
Exp.3 = matrix(c(1,0.5,0.5,2,1,6,2,0.16,1),3,3)
Exp.4 = matrix(c(1,3,4,0.3,1,0.5,0.25,0.3,1),3,3)
Exp.5 = matrix(c(1,4,5,0.25,1,1,0.2,1,1),3,3)
list = list(Exp.1,Exp.2,Exp.3,Exp.4,Exp.5)
# Alternatives' original scores
Alt.1 = c(30,86,-5)
Alt.2 = c(26,77,-12)
Alt.3 = c(22,93,-4)
Alt.4 = c(34,65,-10)
Alt.5 = c(31,80,-8)
Alt.6 = c(29,79,-9)
Alt.7 = c(37,55,-15)
Alt.8 = c(21,69,-11)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8) # Decision matrix
CPP.AHP.Unif(n,list,x)
```

CPP.Axes.Beta *CPP by axes using Beta PERT distributions*

Description

This function computes the CPP by axes, using Beta PERT distributions to randomize the decision matrix. The CPP by axes is used to rank alternatives in multicriteria decision problems. The "Progressive-Conservative" and the "Optimist-Pessimist" axes emulate four decision maker's points of view.

Usage

```
CPP.Axes.Beta(x, s)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

PMax are the joint probabilities of each alternative being higher than the others, per criterion. PMin are the joint probabilities of each alternative being lower than the others, also by criterion. Axes returns the alternatives' scores by axis and ranking for decisionmaking.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Garcia, Pauli A. A. & Sant'Anna, Annibal P. (2015). Vendor and logistics provider selection in the construction sector: A probabilistic preferences composition approach. Pesquisa Operacional 35.2: 363-375.

Examples

```
# Alternatives' original scores
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5) # Decision matrix
s = 4 # Shape
CPP.Axes.Beta(x,s)
```

CPP.Axes.Normal *CPP by axes using Normal distributions*

Description

This function computes the CPP by axes, using Normal distributions to randomize the decision matrix. The CPP by axes is used to rank alternatives in multicriteria decision problems. The "Progressive-Conservative" and the "Optimist-Pessimist" axes emulate four decision maker's points of view.

Usage

```
CPP.Axes.Normal(x)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.

Value

PMax are the joint probabilities of each alternative being higher than the others, per criterion. PMin are the joint probabilities of each alternative being lower than the others, also by criterion. Axes returns the alternatives' scores by axis and ranking for decisionmaking.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Garcia, Pauli A. A. & Sant'Anna, Annibal P. (2015). Vendor and logistics provider selection in the construction sector: A probabilistic preferences composition approach. Pesquisa Operacional 35.2: 363-375.

Examples

```
# Alternatives' original scores
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5) # Decision matrix
CPP.Axes.Normal(x)
```

CPP.Choquet.Beta *CPP by Choquet integrals, using Beta PERT distributions*

Description

This function computes the CPP by Choquet integrals, using Beta PERT distributions to randomize the decision matrix. The CPP by Choquet integrals is used to rank alternatives in multicriteria decision problems.

Usage

```
CPP.Choquet.Beta(x, s)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data elicited from experts.

Value

PMax are the joint probabilities of each alternative being higher than the others, per criterion. Capacities are the interactions of all combined criteria, computed by the Progressive-Optimistic (PO) point of view. Choq returns the alternatives' scores by Choquet integrals and their respetive rankings for decisionmaking. Shap returns the Shapley indices, which are associated with criteria weights.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Examples

```
# Alternatives' original scores
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5) # Decision matrix
s = 4 # Shape
CPP.Choquet.Beta(x,s)
```

CPP.Gini *CPP by the Gini Index, using Beta PERT distributions*

Description

The CPP by the Gini Index is used to rank alternatives by evenness of evaluations, in multicriteria decision problems.

Usage

```
CPP.Gini(x, s)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
s	Shape of Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis.

Value

PMax are the joint probabilities of each alternative being higher than the others, per criterion. CPP.Gini returns the alternatives' scores by the Gini Index and their respective preference ranks for decisionmaking.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer

Gaviao, Luiz O. & Lima, Gilson B.A. (2017) Support decision to player selection: an application of the CPP in soccer, Novas Edições Acadêmicas [in Portuguese].

Examples

```
# Alternatives' original scores
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5) # Decision matrix
s = 4 # Shape
CPP.Gini(x,s)
```

CPP.Malmquist.Beta *CPP by the Malmquist Index, using Beta PERT distributions*

Description

The CPP-Malmquist is used to dynamic evaluation of alternatives, in multicriteria problems, considering two different moments.

Usage

```
CPP.Malmquist.Beta(m1, m2, s)
```

Arguments

m1	Decision matrix of Alternatives (rows) and Criteria (columns) in moment '1'. Benefit criteria must be positive and cost criteria must be negative.
m2	Decision matrix of Alternatives (rows) and Criteria (columns) in the following moment '2'. Benefit criteria must be positive and cost criteria must be negative.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

MC gives the Malmquist Conservative index. MP gives the Malmquist Progressive index. Finally, Index gives the CPP-Malmquist of all alternatives and their rankings for decisionmaking. The indices greater than one represent a relative evolution of the alternative between the two periods, while the indices lower than one reveal the alternatives that decreased performance in relation to the others.

Examples

```
# Alternatives' original scores
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
m1 = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5) # Decision matrix of the previous moment '1'.
Alt.1 = c(3,29,82,-3)
Alt.2 = c(6,28,70,-8)
Alt.3 = c(2,20,99,-8)
Alt.4 = c(5,31,62,-14)
Alt.5 = c(9,27,73,-5)
m2 = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5) # Decision matrix of the following moment '2'.
s = 4 # Shape
CPP.Malmquist.Beta(m1,m2,s)
```

`CPP.mb` *CPP with multiple perspectives for decision-making, based on the 'Moneyball' principle.*

Description

The algorithm evaluates alternatives by integrating the CPP-Tri, the CPP-Malmquist, the CPP-Gini, the alternatives' market values and the CPP by axes. The CPP-mb was originally applied in sports science to evaluate players' performance.

Usage

```
CPP.mb(t1, t2, m, q, s)
```

Arguments

`t1`	Decision matrix of Alternatives (rows) and Criteria (columns) in the moment '1'. Benefit criteria must be positive and cost criteria negative.
`t2`	Decision matrix of Alternatives (rows) and Criteria (columns) in the following moment '2'. Benefit criteria must be positive and cost criteria negative.
`m`	Vector of alternatives' market values.
`q`	Vector of quantiles, indicating the classes' profiles.
`s`	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

Class assigns the alternatives to classes, defined by the indicated profiles. The list of classes also shows the decision matrices to be modeled by CPP-PP. CPP-mb indicates the final scores per class.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Lewis, Michael. (2004) Moneyball: The art of winning an unfair game. WW Norton & Company.

Gaviao, Luiz O. & Lima, Gilson B.A. (2017) Support decision to player selection: an application of the CPP in soccer, Novas Edições Acadêmicas [in Portuguese].

Examples

```
## Decision matrix of the previous moment '1'.
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
Alt.6 = c(6,29,79,-9)
Alt.7 = c(8,37,55,-15)
Alt.8 = c(10,21,69,-11)
t1 = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8)
## Decision matrix of the following moment '2'.
Alt.1 = c(3,29,82,-3)
Alt.2 = c(6,28,70,-8)
Alt.3 = c(2,20,99,-8)
Alt.4 = c(5,31,62,-14)
Alt.5 = c(9,27,73,-5)
Alt.6 = c(4,33,85,-13)
Alt.7 = c(9,39,59,-10)
Alt.8 = c(8,19,77,-9)
t2 = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8)
m = c(100,120,150,140,90,70,110,130) # Market values
q = c(0.65,0.35) # quantiles of class profiles
s = 4 # Shape
CPP.mb(t1,t2,m,q,s)
```

`CPP.rh` *CPP with multiple perspectives for human resources evaluation*

Description

This function computes the CPP-rh, using Beta PERT distributions to randomize the decision matrices. The CPP-rh is used to evaluate alternatives by integrating the CPP-Tri, the CPP-Malmquist, the CPP-Gini and the CPP by axes. The CPP-rh and the CPP-mb are very similar, but the CPP-rh does not include the alternatives's market value.

Usage

```
CPP.rh(t1, t2, q, s)
```

Arguments

t1	Decision matrix of Alternatives (rows) and Criteria (columns) in the previous moment '1'. Benefit criteria must be positive and cost criteria must be negative.
t2	Decision matrix of Alternatives (rows) and Criteria (columns) in the following moment '2'. Benefit criteria must be positive and cost criteria must be negative.
q	Vector of quantiles, indicating the classes' profiles.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

Class identifies the alternatives' classes, according to the selected profiles. CPP-RH returns the alternatives' scores per class.

Examples

```
## Decision matrix of the previous moment '1'.
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
Alt.6 = c(6,29,79,-9)
Alt.7 = c(8,37,55,-15)
Alt.8 = c(10,21,69,-11)
t1 = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8)
## Decision matrix of the following moment '2'.
Alt.1 = c(3,29,82,-3)
Alt.2 = c(6,28,70,-8)
Alt.3 = c(2,20,99,-8)
Alt.4 = c(5,31,62,-14)
Alt.5 = c(9,27,73,-5)
Alt.6 = c(4,33,85,-13)
Alt.7 = c(9,39,59,-10)
Alt.8 = c(8,19,77,-9)
t2 = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8)
q = c(0.65,0.35) # quantiles of class profiles
s = 4 # Shape
CPP.rh(t1,t2,q,s)
```

CPP.SAW *CPP by weighted sum, with weights informed by the user*

Description

This function computes the CPP-SAW, using Normal distributions and weights defined by the decision maker. The CPP-SAW is used to evaluate alternatives by weighted sum.

Usage

```
CPP.SAW(x, w)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
w	Vector of weights assigned by the decision maker. Weights are normalized, just in case their sum differs from the unity.

Value

Weights repeat the parameter 'w' if sum the unity, otherwise are normalized. PMax indicates the joint probabilities of each alternative being higher than the others, per criterion. CPP returns the alternatives' scores by weighted sum, indicating the preference ranks for decisionmaking.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Examples

```
# Decision matrix
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
Alt.6 = c(6,29,79,-9)
Alt.7 = c(8,37,55,-15)
Alt.8 = c(10,21,69,-11)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8)
w = c(0.2,0.3,0.4,0.1)
CPP.SAW(x,w)
```

CPP.SAW.Entropy *CPP by weighted sum, with weights computed from Shannon entropy.*

Description

This function computes the CPP-SAW, using Normal distributions to randomize the decision matrix and weights defined by entropy. The CPP-SAW Entropy is used to evaluate alternatives by weighted sum.

Usage

```
CPP.SAW.Entropy(x)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.

Value

Weights by entropy.PMax are the joint probabilities of each alternative being higher than the others, per criterion. CPP returns the alternatives' scores by weighted sum, indicating the preference ranks for decisionmaking.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Examples

```
## Decision matrix.
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
Alt.6 = c(6,29,79,-9)
Alt.7 = c(8,37,55,-15)
Alt.8 = c(10,21,69,-11)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8)
CPP.SAW.Entropy(x)
```

CPP.Tri.Beta *CPP for sorting alternatives in ordinal classes*

Description

This function computes the CPP-Tri, using Beta PERT distributions to randomize the decision matrix. The CPP-Tri is used to classify alternatives, indicating the order of classes, whose quantity is defined by the decision maker. The probabilities of each alternative being higher and lower than the classes' profiles are composed by the Progressive-Pessimist (PP) point of view.

Usage

```
CPP.Tri.Beta(x, q, s)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
q	Vector of quantiles, indicating the classes' profiles.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

Prob.Plus are the probabilities of each alternative being higher than the classes'profiles. Prob.Minus are the probabilities of each alternative being lower than the classes'profiles. CPP.Tri returns the alternatives' classes.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Sant'Anna, Annibal P.; Costa, Helder G.; Pereira, Valdecy (2015). CPP-TRI: a sorting method based on the probabilistic composition of preferences. International Journal of Information and Decision Sciences 7.3, 193-212.

Examples

```
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
Alt.6 = c(6,29,79,-9)
Alt.7 = c(8,37,55,-15)
Alt.8 = c(10,21,69,-11)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8) # Decision matrix.
q = c(0.65,0.35) # quantiles of classes' profiles.
s = 4 # Shape
CPP.Tri.Beta(x,q,s)
```

CPP.Tri.Choquet *CPP for sorting alternatives, based on Choquet integrals*

Description

This function computes the CPP-Tri with Choquet integrals, using Beta PERT distributions to randomize the decision matrix. The CPP Tri is used to classify alternatives, indicating the order of a number of classes defined by the decision maker. The probabilities of each alternative being higher and lower than the classes' profiles are composed by Choquet integrals.

Usage

```
CPP.Tri.Choquet(x, q, s)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria negative.
q	Vector of quantiles, indicating the classes' profiles.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

Choquet.O returns the probabilities of each alternative being higher than the classes' profiles, composed by Choquet integrals. Choquet.U indicates the probabilities of each alternative being lower than the classes' profiles. CPP.Tri.Chq returns the alternatives' classes.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Sant'Anna, Annibal P.; Lima, Gilson B. A.; Gavião, Luiz O. (2018) A Probabilistic approach to the inequality adjustment of the Human Development Index. Pesquisa Operacional 38.1: 99-116.

Examples

```
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
Alt.6 = c(6,29,79,-9)
Alt.7 = c(8,37,55,-15)
Alt.8 = c(10,21,69,-11)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8) # Decision matrix.
q = c(0.65,0.35) # quantiles of classes' profiles.
s = 4 # Shape
CPP.Tri.Choquet(x,q,s)
```

Entrop.weights *Weights by entropy*

Description

This function computes weights by Shannon's entropy.

Usage

```
Entrop.weights(x)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria negative.

Value

Weights for each criterion.

References

Pomerol, Jean-Charles & Barba-Romero, Sergio. (2012) Multicriterion Decision in Management: Principles and Practice, Springer.

Examples

```
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
Alt.6 = c(6,29,79,-9)
Alt.7 = c(8,37,55,-15)
Alt.8 = c(10,21,69,-11)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5,Alt.6,Alt.7,Alt.8) # Decision matrix.
Entrop.weights(x)
```

PMax.Beta *Probabilities of maximization, by Beta PERT distributions*

Description

This function computes the probabilities of each alternative maximizing the preference per criterion, using Beta PERT distributions to randomize the decision matrix.

Usage

```
PMax.Beta(x, s)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

PMax are the joint probabilities of each alternative being higher than the others, per criterion.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Examples

```
# Decision matrix
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5)
s = 4 # Shape
PMax.Beta(x,s)
```

PMax.Normal *Probabilities of maximization, by Normal distributions*

Description

This function computes the probabilities of each alternative maximizing the preference per criterion, using Normal distributions to randomize the decision matrix.

Usage

```
PMax.Normal(x)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.

Value

PMax are the joint probabilities of each alternative being higher than the others, per criterion.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Examples

```
# Decision matrix
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5)
PMax.Normal(x)
```

PMin.Beta *Probabilities of minimization, by Beta PERT distributions*

Description

This function computes the Probabilities of each alternative minimizing the preference per criterion, using Beta PERT distributions to randomize the decision matrix.

Usage

```
PMin.Beta(x, s)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.
s	Shape of a Beta PERT distribution, as described in the package 'mc2d'. There is no default value, however the higher the shape the higher the kurtosis, which emulates the precision of data.

Value

PMin are the joint probabilities of each alternative being lower than the others, per criterion.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Examples

```
# Decision matrix
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5)
s = 4 # Shape
PMin.Beta(x,s)
```

PMin.Normal *Probabilities of minimization, by Normal distributions*

Description

This function computes the probabilities of each alternative minimizing the preference per criterion, using Normal distributions to randomize the decision matrix.

Usage

```
PMin.Normal(x)
```

Arguments

x	Decision matrix of Alternatives (rows) and Criteria (columns). Benefit criteria must be positive and cost criteria must be negative.

Value

PMin are the joint probabilities of each alternative being lower than the others, per criterion.

References

Sant'Anna, Annibal P. (2015). Probabilistic Composition of Preferences: Theory and Applications, Springer.

Examples

```
# Decision matrix
Alt.1 = c(2,30,86,-5)
Alt.2 = c(4,26,77,-12)
Alt.3 = c(3,22,93,-4)
Alt.4 = c(6,34,65,-10)
Alt.5 = c(5,31,80,-8)
x = rbind(Alt.1,Alt.2,Alt.3,Alt.4,Alt.5)
PMin.Normal(x)
```

Index