Package ‘ConsRank’

April 29, 2017

Type Package

Title Compute the Median Ranking(s) According to the Kemeny's Axiomatic Approach

Version 2.0.1

Date 2017-04-28

Author Antonio D'Ambrosio <antdambr@unina.it>, Sonia Amodio <sonia.amodio@unina.it>, Giulio Mazzeo <giuliomazzeo@gmail.com>

Maintainer Antonio D'Ambrosio <antdambr@unina.it>

Depends proxy,rgl,gtools

Description Compute the median ranking according to the Kemeny's axiomatic approach.
Rankings can or cannot contain ties, rankings can be both complete or incomplete.
The package contains both branch-and-
bound algorithms and heuristic solutions recently proposed.
The package also provide some useful utilities for deal with preference rankings.
Essential references:
Emond, E.J., and Mason, D.W. (2002) <doi:10.1002/mcda.313>;
D'Ambrosio, A., Amodio, S., and Iorio, C. (2015) <doi:10.1285/i20705948v8n2p198>;
Amodio, S., D'Ambrosio, A., and Siciliano R. (2016) <doi:10.1016/j.ejor.2015.08.048>;
D'Ambrosio, A., Mazzeo, G., Iorio, C., and Siciliano, R. (2017) <doi:10.1016/j.cor.2017.01.017>.

License GPL-3

URL https://www.r-project.org/

Repository CRAN

RoxygenNote 6.0.1

NeedsCompilation no

Date/Publication 2017-04-28 22:24:06 UTC

R topics documented:

`ConsRank-package` *Compute the Median Ranking According to the Kemeny's Axiomatic Approach*

Description

Compute the median ranking according to the Kemeny's axiomatic approach. Rankings can or cannot contain ties, rankings can be both complete or incomplete. The package contains both branch-and-bound and heuristic solutions

Details

Package:	ConsRank
Type:	Package
Version:	2.0.1
Date:	2017-04-28
License:	GPL-3

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>, Sonia Amdio <sonia.amodio@unina.it>, Giulio Mazzeo <giuliomazzeo@gmail.com>

Maintainer: Antonio D'Ambrosio <antdambr@unina.it>

References

Kemeny, J. G., & Snell, J. L. (1962). Mathematical models in the social sciences (Vol. 9). New York: Ginn.

Marden, J. I. (1996). Analyzing and modeling rank data. CRC Press.

Emond, E. J., & Mason, D. W. (2002). A new rank correlation coefficient with application to the consensus ranking problem. Journal of Multi-Criteria Decision Analysis, 11(1), 17-28.

D'Ambrosio, A. (2008). Tree based methods for data editing and preference rankings. Ph.D. thesis. http://www.fedoa.unina.it/id/eprint/2746

Heiser, W. J., & D'Ambrosio, A. (2013). Clustering and prediction of rankings within a Kemeny distance framework. In Algorithms from and for Nature and Life (pp. 19-31). Springer International Publishing.

Amodio, S., D'Ambrosio, A. & Siciliano, R (2016). Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach. European Journal of Operational Research, vol. 249(2).

D'Ambrosio, A., Amodio, S. & Iorio, C. (2015). Two algorithms for finding optimal solutions of the Kemeny rank aggregation problem for full rankings. Electronic Journal of Applied Statistical Analysis, vol. 8(2).

D'Ambrosio, A., Mazzeo, G., Iorio, C., & Siciliano, R. (2017). A differential evolution algorithm for finding the median ranking under the Kemeny axiomatic approach. Computers & Operations Research, vol. 82.

Examples

```
## load APA data set, full version
data(APAFULL)
## Emond and Mason Branch-and-Bound algorithm.
## If the number of object is higher than 20, EMCons function may work for several minutes.
## Use either QuickCons, DECOR, FASTcons or FASTDECOR instead
CR=EMCons(APAFULL)
```

```
TR=tabulaterows(APAFULL)
CR2=FASTcons(TR$X,TR$Wk,maxiter=5)
CR3=QuickCons(TR$X,TR$Wk)
CR4=DECOR(TR$X,TR$Wk)

#####################################
### load sports data set
#data(sports)
### FAST algorithm
#CR=FASTcons(sports,maxiter=10)
#####################################

#######################################
### load Emond and Mason data set
#data(EMD)
### matrix X contains rankings
#X=EMD[,1:15]
### vector Wk contains frequencies
#Wk=EMD[,16]
### QUICK algorithm
#CR=QuickCons(X,Wk)
#######################################
```

APAFULL *American Psychological Association dataset, full version*

Description

The American Psychological Association dataset includes 15449 ballots of the election of the president in 1980, 5738 of which are complete rankings, in which the candidates are ranked from most to least favorite.

Usage

```
data(APAFULL)
```

Source

Diaconis, P. (1988). Group representations in probability and statistics. Lecture Notes-Monograph Series, i-192., pag. 96.

APAred — *American Psychological Association dataset, reduced version with only full rankings*

Description

The American Psychological Association reduced dataset includes 5738 ballots of the election of the president in 1980, in which the candidates are ranked from most to least favorite.

Usage

```
data(APAred)
```

Source

Diaconis, P. (1988). Group representations in probability and statistics. Lecture Notes-Monograph Series, i-192., pag. 96.

BBconsensus — *Find the first approximation to the consensus ranking. Most of the time the output is a solution, maybe not unique*

Description

Find a first approximation to the consensus ranking.

Usage

```
BBconsensus(RR, cij, FULL = FALSE, PS = FALSE)
```

Arguments

`RR`	Candidate to be the consensus ranking
`cij`	Combined input matrix of the data set
`FULL`	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of complete rankings. FULL=TRUE if the function is called by BBFULL algorithm.
`PS`	Default PS=FALSE. If PS=TRUE the number of evaluated branches is diplayed

Value

a "list" containing the following components:

cons	a first approximation of the median ranking
pen	penalty value

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

Amodio, S., D'Ambrosio, A. and Siciliano, R. (2016). Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach. European Journal of Operational Research, 249(2), 667-676.

BBconsensus2 — *Core function in computing consensus ranking as defined by Emond and Mason (2002)*

Description

Core function in computing consensus ranking as defined by Emond and Mason (2002), recalled by EMCons function

Usage

```
BBconsensus2(RR, cij, Po, PS = TRUE, FULL = FALSE)
```

Arguments

`RR`	A ranking
`cij`	combined input matrix
`Po`	current penalty
`PS`	If PS=true, it prints the evaluating branches
`FULL`	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of complete rankings. FULL=TRUE if the function is called by BBFULL algorithm.

Value

median ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

Emond, E. J., and Mason, D. W. (2002). A new rank correlation coefficient with application to the consensus ranking problem. Journal of Multi-Criteria Decision Analysis, 11(1), 17-28. # D'Ambrosio, A., Amodio, S., and Iorio, C. (2015). Two algorithms for finding optimal solutions of the Kemeny rank aggregation problem for full rankings. Electronic Journal of Applied Statistical Analysis, 8(2), 198-213.

See Also

`EMCons` Emond and Mason branch-and-bound algorithm

`BBFULL` D'Ambrosio et al. branch-and-bound algorithm for full rankings

---

BBFULL — *Branch-and-Bound algorithm to find the median ranking in the space of full (or complete) rankings.*

Description

Branch-and-bound algorithm to find consensus ranking as definned by D'Ambrosio et al. (2015). If the number of objects to be ranked is large (greater than 20 or 25), it can work for very long time. Use either QuickCons or FASTcons with the option FULL=TRUE instead

Usage

```
BBFULL(X, Wk = NULL, PS = TRUE)
```

Arguments

`X`	A N by M data matrix, in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. The data matrix can contain both full and tied rankings, or incomplete rankings. Alternatively X can contain the rankings observed only once. In this case the argument Wk must be used
`Wk`	Optional: the frequency of each ranking in the data
`PS`	If PS=TRUE, on the screen some information about how many branches are processed are displayed

Details

If the objects to be ranked is large (>25 - 30), it can take long time to find the solutions

Value

a "list" containing the following components:

Consensus	the Consensus Ranking
Tau	averaged TauX rank correlation coefficient
Eltime	Elapsed time in seconds

Author(s)

Antonio D'Ambrosio `<antdambr@unina.it>`

References

D'Ambrosio, A., Amodio, S., and Iorio, C. (2015). Two algorithms for finding optimal solutions of the Kemeny rank aggregation problem for full rankings. Electronic Journal of Applied Statistical Analysis, 8(2), 198-213.

See Also

`FASTcons` FAST algorithm algorithm.

`QuickCons` Quick algorithm.

Examples

```
data(APAFULL)
CR=BBFULL(APAFULL)
```

---

branches *Auxliary code recalled by other routines*

Description

Branches discovery

Usage

```
branches(brR, cij, b, Po, ord, Pb, FULL = FALSE)
```

Arguments

`brR`	Current processed branche of the BB algorithm
`cij`	Combined input matrix
`b`	Other inputs recalled by main functions
`Po`	Other inputs recalled by main functions
`ord`	Other inputs recalled by main functions
`Pb`	Other inputs recalled by main functions
`FULL`	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of complete rankings. TRUE=TRUE if the function is called by BBFULL algorithm.

Value

a "list" containing the following components:

cR	ranking belonging to the branche
pcR	penalty of the current ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

`BU` *Brook and Upton data*

Description

The data consist of ballots of three candidates, where the 948 voters rank the candidates from 1 to 3. Data are in form of frequency table.

Usage

```
data(BU)
```

Source

Brook, D., & Upton, G. J. G. (1974). Biases in local government elections due to position on the ballot paper. Applied Statistics, 414-419.

References

Marden, J. I. (1996). Analyzing and modeling rank data. CRC Press, pag. 153.

Examples

```
data(BU)
polyplot(BU[,1:3],Wk=BU[,4])
```

`childclosint` *Transform the vector into ranking for DECOR*

Description

Closest integer approach is used: the elements of x are rounded to the closest integer. Then check if any solution exists outside of the bounds (and get it back inside the bounds randomly). Finally repair the solution if repetitions exist.

Usage

```
childclosint(r)
```

Arguments

`r`	mutated vector

Value

a valid ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo <giuliomazzeo@gmail.com>

References

Davendra, D., and Onwubolu, G. (2007). Enhanced differential evolution hybrid scatter search for discrete optimization. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (pp. 1156-1162). IEEE.

---

childtie *Auxiliary function*

Description

Apply discretization and convert to rank

Usage

```
childtie(r)
```

Arguments

r	a candidate rank vector

Value

a ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo <giuliomazzeo@gmail.com>

`combincost` *Auxiliary function called by DECORcore*

Description

Calculates the sum of distances between a candidate ranking and the data set of rankings

Usage

```
combincost(ranking, cij, M)
```

Arguments

`ranking`	the candidate ranking
`cij`	combined input matrix
`M`	number of judges

Value

a "list" containing the following components:

tp	tauX associated to the ranking
cp	cost associated to the ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo

`combinpmatr` *Combined input matrix of a data set*

Description

Compute the Combined input matrix of a data set as defined by Emond and Mason (2002)

Usage

```
combinpmatr(X, Wk = NULL)
```

Arguments

`X`	A data matrix N by M, in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. Alternatively X can contain the rankings observed only once. In this case the argument Wk must be used
`Wk`	Optional: the frequency of each ranking in the data

Value

The M by M combined input matrix

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

Emond, E. J., and Mason, D. W. (2002). A new rank correlation coefficient with application to the consensus ranking problem. Journal of Multi-Criteria Decision Analysis, 11(1), 17-28.

See Also

tabulaterows frequency distribution of a ranking data.

Examples

```
data(APAred)
CI=combinpmatr(APAred)
TR=tabulaterows(APAred)
CI=combinpmatr(TR$X,TR$Wk)
```

---

crossover *Apply the (binomial) crossover for DE algorithm*

Description

Binomial crossover stipulates that crossover will occur on each of the D values in a solution whenever a randomly generated number between 0 and 1 is within the CR range.

Usage

```
crossover(x, v, CR)
```

Arguments

x	target ranking
v	donor individual (mutaded x)
CR	Crossover range

Value

modified ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo <giuliomazzeo@gmail.com>

DECOR *Differential Evolution algorithm for Median Ranking*

Description

Differential evolution algorithm for median ranking detection. It works with full, tied and partial rankings. The solution con be constrained to be a full ranking or a tied ranking

Usage

```
DECOR(X, Wk = NULL, NP = 15, L = 100, FF = 0.4, CR = 0.9,
  FULL = FALSE)
```

Arguments

`X`	A N by M data matrix, in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. Alternatively X can contain the rankings observed only once. In this case the argument Wk must be used
`Wk`	Optional: the frequency of each ranking in the data
`NP`	The number of population individuals
`L`	Generations limit: maximum number of consecutive generations without improvement
`FF`	The scaling rate for mutation. Must be in [0,1]
`CR`	The crossover range. Must be in [0,1]
`FULL`	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of full rankings.

Details

It works with a very large number of items to be ranked. Empirically, the number of population individuals (the NP parameter) can be set equal to 10, 20 or 30 for problems till 20, 50 and 100 items. Both scaling rate and crossover ratio (parameters FF and CR) must be set by the user. The default options (FF=0.4, CR=0.9) work well for a large variety of data sets

Value

a "list" containing the following components:

Consensus	the Consensus Ranking
Tau	averaged TauX rank correlation coefficient
Eltime	Elapsed time in seconds

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo <giuliomazzeo@gmail.com>

References

D'Ambrosio, A., Mazzeo, G., Iorio, C., and Siciliano, R. (2017). A differential evolution algorithm for finding the median ranking under the Kemeny axiomatic approach. Computers and Operations Research, vol. 82, pp. 126-138.

See Also

`FASTcons` FAST algorithm.

`QuickCons` Quick algorithm.

`EMCons` Branch-and-bound algorithm.

Examples

```
data(EMD)
CR=DECOR(EMD[,1:15],EMD[,16])
```

DECORcore — *Differential Evolution algorithm for Median Ranking*

Description

Core function of the DECOR algorithm

Usage

```
DECORcore(cij, NJ, NP = 15, L = 50, FF = 0.4, CR = 0.9, FULL = FALSE)
```

Arguments

`cij`	combined input matrix
`NJ`	the number of judjes
`NP`	The number of population individuals
`L`	Generations limit: maximum number of consecutive generations without improvement
`FF`	The scaling rate for mutation. Must be in [0,1]
`CR`	The crossover range. Must be in [0,1]
`FULL`	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of full rankings. In this case, the data matrix must contain full rankings.

Value

a "list" containing the following components:

ConsR	the Consensus Ranking
Tau	averaged TauX rank correlation coefficient
besti	matrix of best individuals for every generation
bestc	vector of best individuals' cost for every gen
bests	
vector of best individuals avgTau	
maximum average tauX Eltime	Elapsed time in seconds

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo <giuliomazzeo@gmail.com>

References

D'Ambrosio, A., Mazzeo, G., Iorio, C., and Siciliano, R. (2017). A differential evolution algorithm for finding the median ranking under the Kemeny axiomatic approach. Computers and Operations Research, vol. 82, pp. 126-138.

EMCons — *Branch-and-bound algorithm to find consensus (median) ranking according to the Kemeny's axiomatic approach*

Description

Branch-and-bound algorithm to find consensus ranking as definned by Emond and Mason (2002). If the number of objects to be ranked is large (greater than 15 or 20, specially if there are missing rankings), it can work for very long time.

Usage

```
EMCons(X, Wk = NULL, PS = TRUE)
```

Arguments

`X`	A N by M data matrix, in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. Alternatively X can contain the rankings observed only once. In this case the argument Wk must be used
`Wk`	Optional: the frequency of each ranking in the data
`PS`	If PS=TRUE, on the screen some information about how many branches are processed are displayed

Details

If the objects to be ranked is large (>15-20) with some missing, it can take long time to find the solutions. If the searching space is limited to the space of full rankings (also incomplete rankings, but without ties), use the function BBFULL or the functions FASTcons and QuickCons with the option FULL=TRUE.

Value

a "list" containing the following components:

Consensus	the Consensus Ranking
Tau	averaged TauX rank correlation coefficient
Eltime	Elapsed time in seconds

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

Emond, E. J., and Mason, D. W. (2002). A new rank correlation coefficient with application to the consensus ranking problem. Journal of Multi-Criteria Decision Analysis, 11(1), 17-28.

See Also

FASTcons FAST algorithm algorithm.

QuickCons Quick algorithm.

BBFULL Branc-and-bound algorithm for full rankings.

Examples

```
data(Idea)
RevIdea=6-Idea
# as 5 means "most associated", it is necessary compute the reverse ranking of
# each rankings to have rank 1 = "most associated" and rank 5 = "least associated"
CR=EMCons(RevIdea)
```

EMD *Emond and Mason data*

Description

Data simuated by Emond and Mason to check their branch-and-bound algorithm. There are 112 voters ranking 15 objects. There are 21 uncomplete rankings. Data are in form of frequency table.

Usage

```
data(EMD)
```

Source

Emond, E. J., & Mason, D. W. (2000). A new technique for high level decision support. Department of National Defence, Operational Research Division, pag. 28.

References

Emond, E. J., & Mason, D. W. (2000). A new technique for high level decision support. Department of National Defence, Operational Research Division, pag. 28.

Examples

```
data(EMD)
CR=QuickCons(EMD[,1:15],EMD[,16])
```

FASTcons — *FAST algorithm to find consensus (median) ranking. FAST algorithm to find consensus (median) ranking defined by Amodio, D'Ambrosio and Siciliano (2016). It returns at least one of the solutions. If there are multiple solutions, sometimes it returns all the solutions, sometimes it returns some solutions, always it returns at least one solution.*

Description

FAST algorithm to find consensus (median) ranking.

FAST algorithm to find consensus (median) ranking defined by Amodio, D'Ambrosio and Siciliano (2016). It returns at least one of the solutions. If there are multiple solutions, sometimes it returns all the solutions, sometimes it returns some solutions, always it returns at least one solution.

Usage

```
FASTcons(X, Wk = NULL, maxiter = 50, FULL = FALSE, PS = FALSE)
```

Arguments

`X`	is a ranking data matrix
`Wk`	is a vector of weights
`maxiter`	maximum number of iterations: default = 50.
`FULL`	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of full rankings.
`PS`	Default PS=FALSE. If PS=TRUE the number of current iteration is diplayed

Value

a "list" containing the following components:

Consensus	the Consensus Ranking
Tau	averaged TauX rank correlation coefficient
Eltime	Elapsed time in seconds

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Sonia Amodio <sonia.amodio@unina.it>

References

Amodio, S., D'Ambrosio, A. and Siciliano, R. (2016). Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach. European Journal of Operational Research, 249(2), 667-676.

See Also

EMCons Emond and Mason branch-and-bound algorithm.

QuickCons Quick algorithm.

Examples

```
##data(EMD)
##X=EMD[,1:15]
##Wk=matrix(EMD[,16],nrow=nrow(X))
##CR=FASTcons(X,Wk,maxiter=100)
##These lines produce all the three solutions in less than a minute.

data(sports)
CR=FASTcons(sports,maxiter=10)
```

FASTDECOR — *FAST algorithm calling DECOR*

Description

FAST algorithm repeats DECOR a prespecified number of time. It returns the best solutions among the iterations

Usage

```
FASTDECOR(X, Wk = NULL, maxiter = 10, NP = 15, L = 100, FF = 0.4,
  CR = 0.9, FULL = FALSE, PS = TRUE)
```

Arguments

X	A N by M data matrix, in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. Alternatively X can contain the rankings observed only once. In this case the argument Wk must be used
Wk	Optional: the frequency of each ranking in the data
maxiter	maximum number of iterations. Default 10
NP	The number of population individuals
L	Generations limit: maximum number of consecutive generations without improvement
FF	The scaling rate for mutation. Must be in [0,1]
CR	The crossover range. Must be in [0,1]
FULL	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of full rankings. In this case, the data matrix must contain full rankings.
PS	Default PS=TRUE. If PS=TRUE the number of a multiple of 5 iterations is diplayed

Value

a "list" containing the following components:

Consensus	the Consensus Ranking
Tau	averaged TauX rank correlation coefficient
Eltime	Elapsed time in seconds

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo <giuliomazzeo@gmail.com>

References

D'Ambrosio, A., Mazzeo, G., Iorio, C., and Siciliano, R. (2017). A differential evolution algorithm for finding the median ranking under the Kemeny axiomatic approach. Computers and Operations Research, vol. 82, pp. 126-138.

Examples

```
#data(EMD)
#CR=FASTDECOR(EMD[,1:15],EMD[,16])
```

findbranches *Auxiliary function*

Description

Find correct branches in the Branch-and-Bound algorithms

Usage

```
findbranches(R, ord, b, FULL = FALSE)
```

Arguments

R	Candidate to be the consensus ranking
ord	other input values recalled by other routines
b	other input values recalled by other routines
FULL	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of complete rankings. TRUE=TRUE if the function is called by BBFULL algorithm.

Value

a candidate to be the median ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Sonia Amodio <sonia.amodio@unina.it>

findconsensusBB *Find a first ranking candidate to be the median ranking*

Description

Auxiliary function: it finds a first ranking candidate to be the consensus ranking by observing the Combined input matrix

Usage

```
findconsensusBB(cij)
```

Arguments

cij	combined input matrix

Value

candidate median ranking

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

`German` *German political goals*

Description

Ranking data of 2262 German respondents about the desirability of the four political goals: a = the maintenance of order in the nation; b = giving people more say in the decisions of government; c = growthing rising prices; d = protecting freedom of speech

Usage

```
data(German)
```

Source

Croon, M. A. (1989). Latent class models for the analysis of rankings. Advances in psychology, 60, 99-121.

Examples

```
data(German)
TR=tabulaterows(German)
polyplot(TR$X,Wk=TR$Wk,nobj=4)
```

`Idea` *Idea data set*

Description

98 college students where asked to rank five words, (thought, play, theory, dream, attention) regarding its association with the word idea, from 5=most associated to 1=least associated.

Usage

```
data(Idea)
```

Source

Fligner, M. A., & Verducci, J. S. (1986). Distance based ranking models. Journal of the Royal Statistical Society. Series B (Methodological), 359-369.

Examples

```
data(Idea)
revIdea=6-Idea
TR=tabulaterows(revIdea)
CR=QuickCons(TR$X,TR$Wk)
colnames(CR$Consensus)=colnames(Idea)
```

---

kemenyd *Kemeny distance*

Description

Compute the Kemeny distance of a data matrix containing preference rankings, or compute the kemeny distance between two (matrices containing) rankings.

Usage

```
kemenyd(X, Y = NULL)
```

Arguments

X	A N by M data matrix, in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. If there is only X as input, the output is a square distance matrix
Y	A row vector, or a n by M data matrix in which there are n judges and the same M objects as X to be judged.

Value

If there is only X as input, d = square distance matrix. If there is also Y as input, d = matrix with N rows and n columns.

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

Kemeny, J. G., & Snell, L. J. (1962). Preference ranking: an axiomatic approach. Mathematical models in the social sciences, 9-23.

See Also

`Tau_X` TauX rank correlation coefficient

Examples

```
data(Idea)
RevIdea=6-Idea ##as 5 means "most associated", it is necessary compute the reverse
#ranking of each rankings to have rank 1 = "most associated" and rank 5 = "least associated"
KD=kemenyd(RevIdea)
KD2=kemenyd(RevIdea[1:10,],RevIdea[55,])
```

---

kemenydesign *Auxiliary function*

Description

Define a design matrix to compute Kemeny distance

Usage

```
kemenydesign(X)
```

Arguments

X	A N by M data matrix, in which there are N judges and M objects to be judged. Each row is a ranking of the objects represented by the columns.

Value

Design matrix

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

D'Ambrosio, A. (2008). Tree based methods for data editing and preference rankings. Unpublished PhD Thesis. Universita' degli Studi di Napoli Federico II.

kemenyscore *Score matrix according Kemeny (1962)*

Description

Given a ranking, it computes the score matrix as defined by Emond and Mason (2002)

Usage

```
kemenyscore(X)
```

Arguments

X	a ranking (must be a row vector or, better, a matrix with one row and M columns)

Value

the M by M score matrix

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

Kemeny, J and Snell, L. (1962). Mathematical models in the social sciences.

See Also

scorematrix The score matrix as defined by Emond and Mason (2002)

Examples

```
Y = matrix(c(1,3,5,4,2),1,5)
SM=kemenyscore(Y)
#
Z=c(1,2,3,2)
SM2=kemenyscore(Z)
```

`labels` *Transform a ranking into a ordering.*

Description

Given a ranking (or a matrix of rank data), transforms it into an ordering (or a ordering matrix)

Usage

```
labels(x, m, label = 1:m, labs)
```

Arguments

- `x` a ranking, or a n by m data matrix in which there are n judges ranking m objects
- `m` the number of objects
- `label` optional: the name of the objects
- `labs` labs = 1 displays the names of the objects if there is argument "label", otherwise displays the permutation of first m integer. labs = 2 is to be used only if the argument "label" is not defined. In such a case it displays the permutation of the first m letters

Value

the ordering

Author(s)

Sonia Amodio <sonia.amodio@unina.it>

Examples

```
data(Idea)
TR=tabulaterows(Idea)
Ord=labels(TR$X,ncol(Idea),colnames(Idea),labs=1)
Ord2=labels(TR$X,ncol(Idea),labs=2)
cbind(Ord,TR$Wk)
cbind(Ord2,TR$Wk)
```

`mutaterand1` *Mutation phase*

Description

Creates mutation vector v based on the current population X. We use the rand/1/bin system

Usage

```
mutaterand1(X, FF, i)
```

Arguments

`X`	population matrix
`FF`	scaling factor
`i`	population index to be ignored

Value

the mutated vector

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Giulio Mazzeo <giuliomazzeo@gmail.com>

`Penalty` *Auxiliary function*

Description

Assign a penalty to the branches of the FASTcons amd QuickCons algorithms

Usage

```
Penalty(CR, cij, indice)
```

Arguments

`CR`	candidate to be the median ranking
`cij`	combined input matrix
`indice`	other input called by other functions

Value

a penalty value

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Sonia Amodio <sonia.amodio@unina.it>

References

Amodio, S., D'Ambrosio, A. and Siciliano, R. (2016). Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach. European Journal of Operational Research, 249(2), 667-676.

PenaltyBB2 *Auxiliary function*

Description

Assign a penalty to the branches of the BB algorithms

Usage

```
PenaltyBB2(cij, candidate, ord)
```

Arguments

cij	combined input matrix
candidate	candidate to be the median ranking
ord	other input called by other functions

Value

computed penalty

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

polyplot	*Plot rankings on a permutation polytope of 3 o 4 objects containing all possible ties*

Description

Plot rankings a permutation polytope that is the geometrical space of preference rankings. The plot is available for 3 or for 4 objects

Usage

```
polyplot(X = NULL, L = NULL, Wk = NULL, nobj = 3)
```

Arguments

X	the sample of rankings. Most of the time it is returned by tabulaterows
L	labels of the objects
Wk	frequency associated to each ranking
nobj	number of objects. It must be either 3 or 4

Details

polyplot() plots the universe of 3 objecys. polyplot(nobj=4) plots the universe of 4 objecys.

Value

the permutation polytope

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Sonia Amodio <sonia.amodio@unina.it>

References

Thompson, G. L. (1993). Generalized permutation polytopes and exploratory graphical methods for ranked data. The Annals of Statistics, 1401-1430. # Heiser, W. J., and D'Ambrosio, A. (2013). Clustering and prediction of rankings within a Kemeny distance framework. In Algorithms from and for Nature and Life (pp. 19-31). Springer International Publishing.

See Also

tabulaterows frequency distribution for ranking data.

Examples

```
polyplot()
#polyplot(nobj=4)
data(BU)
polyplot(BU[,1:3],Wk=BU[,4])
```

QuickCons — *Quick algorithm to find up to 4 solutions to the consensus ranking problem*

Description

The Quick algorithm finds up to 4 solutions. Solutions reached are most of the time optimal solutions.

Usage

```
QuickCons(X, Wk = NULL, FULL = FALSE, PS = FALSE)
```

Arguments

`X`	A N by M data matrix in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. Alternatively X can contain the rankings observed only once in the sample. In this case the argument Wk must be used
`Wk`	Optional: the frequency of each ranking in the data
`FULL`	Default FULL=FALSE. If FULL=TRUE, the searching is limited to the space of full rankings.
`PS`	Default PS=FALSE. If PS=TRUE the number of evaluated branches is diplayed

Value

a "list" containing the following components:

Consensus	the Consensus Ranking
Tau	averaged TauX rank correlation coefficient
Eltime	Elapsed time in seconds

Author(s)

Antonio D'Ambrosio <antdambr@unina.it> and Sonia Amodio <sonia.amodio@unina.it>

References

Amodio, S., D'Ambrosio, A. and Siciliano, R. (2016). Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach.

European Journal of Operational Research, 249(2), 667-676.

See Also

FASTcons FAST algorithm.

QuickCons Quick algorithm.

Examples

```
data(EMD)
CR=QuickCons(EMD[,1:15],EMD[,16])
```

---

reordering *Given a vector (or a matrix), returns an ordered vector (or a matrix with ordered vectors)*

Description

Given a ranking of M objects (or a matrix with M columns), it reduces it in "natural" form (i.e., with integers from 1 to M)

Usage

```
reordering(X)
```

Arguments

X	a ranking, or a ranking data matrix

Value

a ranking in natural form

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

`ReorderingBB` *Auxiliary function*

Description

It allows to reord the objects to be processed in the BB algorithms

Usage

```
ReorderingBB(RR)
```

Arguments

`RR`	A ranking

Value

A reordered ranking

Author(s)

Sonia Amodio `<sonia.amodio@unina.it>`

`scorematrix` *Score matrix according Emond and Mason (2002)*

Description

Given a ranking, it computes the score matrix as defined by Emond and Mason (2002)

Usage

```
scorematrix(X)
```

Arguments

`X`	a ranking (must be a row vector or, better, a matrix with one row and M columns)

Value

the M by M score matrix

Author(s)

Antonio D'Ambrosio `<antdambr@unina.it>`

References

Emond, E. J., and Mason, D. W. (2002). A new rank correlation coefficient with application to the consensus ranking problem. Journal of Multi-Criteria Decision Analysis, 11(1), 17-28.

See Also

`combinpmatr` The combined inut matrix

Examples

```
Y = matrix(c(1,3,5,4,2),1,5)
SM=scorematrix(Y)
#
Z=c(1,2,4,3)
SM2=scorematrix(Z)
```

`sports` *sports data*

Description

130 students at the University of Illinois ranked seven sports according to their preference (Baseball, Football, Basketball, Tennis, Cycling, Swimming, Jogging).

Usage

```
data(sports)
```

Source

Marden, J. I. (1996). Analyzing and modeling rank data. CRC Press.

Examples

```
data(sports)
```

`tabulaterows` *Frequency distribution of a sample of rankings*

Description

Given a sample of preference rankings, it compute the frequency associated to each ranking

Usage

```
tabulaterows(X, miss = FALSE)
```

Arguments

`X`	a N by M data matrix containing N judges judging M objects
`miss`	TRUE if there are missing data (either partial or incomplete rankings): default: FALSE

Value

a "list" containing the following components:

X	the unique rankings
Wk	the frequency associated to each ranking
tabfreq	frequency table

Author(s)

Sonia Amodio <sonia.amodio@unina.it>

Examples

```
data(Idea)
TR=tabulaterows(Idea)
FR=TR$Wk/sum(TR$Wk)
RF=cbind(TR$X,FR)
colnames(RF)=c(colnames(Idea),"fi")
#compute modal ranking
maxfreq=which(RF[,6]==max(RF[,6]))
labels(RF[maxfreq,1:5],5,colnames(Idea),labs=1)
#
data(APAred)
TR=tabulaterows(APAred)
#
data(APAFULL)
TR=tabulaterows(APAFULL)
CR1=EMCons(TR$X,TR$Wk)
CR2=FASTcons(TR$X,TR$Wk,maxiter=15)
CR3=QuickCons(TR$X,TR$Wk)
```

Tau_X *TauX (tau exstension) rank correlation coefficient*

Description

Tau exstension is a new rank correlation coefficient defined by Emond and Mason (2002)

Usage

```
Tau_X(X, Y = NULL)
```

Arguments

X	a M by N data matrix, in which there are N judges and M objects to be judged. Each row is a ranking of the objects which are represented by the columns. If there is only X as input, the output is a square matrix containing the Tau_X rcc.
Y	A row vector, or a n by M data matrix in which there are n judges and the same M objects as X to be judged.

Value

Tau_x rank correlation coefficient

Author(s)

Antonio D'Ambrosio <antdambr@unina.it>

References

Emond, E. J., and Mason, D. W. (2002). A new rank correlation coefficient with application to the consensus ranking problem. Journal of Multi-Criteria Decision Analysis, 11(1), 17-28.

See Also

kemenyd Kemeny distance

Examples

```
data(BU)
RD=BU[,1:3]
Tau=Tau_X(RD)
Tau1_3=Tau_X(RD[1,],RD[3,])
```

USAranks *USA rank data*

Description

Random subset of the rankings collected by O'Leary Morgan and Morgon (2010) on the 50 American States. The 368 number of items (the number of American States) is equal to 50, and the number of rankings is equal to 104. These data concern rankings of the 50 American States on three particular aspects: socio-demographic characteristics, health care expenditures and crime statistics.

Usage

```
data(USAranks)
```

Source

Amodio, S., D'Ambrosio, A. & Siciliano, R (2015). Accurate algorithms for identifying the median ranking when dealing with weak and partial rankings under the Kemeny axiomatic approach. European Journal of Operational Research. DOI: 10.1016/j.ejor.2015.08.048

References

O'Leary Morgan, K., Morgon, S., (2010). State Rankings 2010: A Statistical view of America; Crime State Ranking 2010: Crime Across America; Health Care State Rankings 2010: Health Care Across America. CQ Press.

Examples

```
data(USAranks)
```

Index