Package 'Rankcluster'

August 28, 2019

Type Package

Title Model-Based Clustering for Multivariate Partial Ranking Data

Version 0.94.1

Date 2019-08-26

Description Implementation of a model-based clustering algorithm for ranking data (C. Biernacki, J. Jacques (2013) <doi:10.1016/j.csda.2012.08.008>). Multivariate rankings as well as partial rankings are taken into account. This algorithm is based on an extension of the Insertion Sorting Rank (ISR) model for ranking data, which is a meaningful and effective model parametrized by a position parameter (the modal ranking, quoted by mu) and a dispersion parameter (quoted by pi). The heterogeneity of the rank population is modelled by a mixture of ISR, whereas conditional independence assumption is considered for multivariate rankings.

License GPL (>= 2)

Copyright Inria - Université de Lille

Depends R (>= 2.10), methods

Imports Rcpp

LinkingTo Rcpp, RcppEigen

Encoding UTF-8

RoxygenNote 6.1.1

Author Quentin Grimonprez [aut, cre],
Julien Jacques [aut],
Christophe Biernacki [aut]

Maintainer Quentin Grimonprez <quentin.grimonprez@inria.fr>

Repository CRAN

Repository/R-Forge/Project rankclust

Repository/R-Forge/Revision 71

Repository/R-Forge/DateTimeStamp 2019-08-27 12:07:18

Date/Publication 2019-08-27 23:40:05 UTC

NeedsCompilation yes

R topics documented:

`Rankcluster-package` *Model-Based Clustering for Multivariate Partial Ranking Data*

Description

This package proposes a model-based clustering algorithm for ranking data. Multivariate rankings as well as partial rankings are taken into account. This algorithm is based on an extension of the Insertion Sorting Rank (ISR) model for ranking data, which is a meaningful and effective model parametrized by a position parameter (the modal ranking, quoted by mu) and a dispersion parameter (quoted by pi). The heterogeneity of the rank population is modelled by a mixture of ISR, whereas conditional independence assumption is considered for multivariate rankings.

Details

The main function is rankclust.

Author(s)

Maintainer: Quentin Grimonprez <quentin.grimonprez@inria.fr>

References

[1] C.Biernacki and J.Jacques (2013), A generative model for rank data based on sorting algorithm, Computational Statistics and Data Analysis, 58, 162-176.

[2] J.Jacques and C.Biernacki (2012), Model-based clustering for multivariate partial ranking data, Inria Research Report n 8113.

Examples

```
# see vignette
# vignette("Rankcluster")

# main function of the package for run the algorithm
data(big4)
result = rankclust(big4$data, K = 2, m = big4$m, Ql = 200, Bl = 100, maxTry = 2)
```

APA *rank data : APA*

Description

This dataset contains the 5738 full rankings resulting from the American Psychological Association (APA) presidential election of 1980. For this election, members of APA had to rank five candidates in order of preference.

For information, a total of 15449 votes have been registred for this election, but only the 5738 full rankings are reported in the APA dataset. Candidates A and C were research psychologists, candidates D and E were clinical psychologists and candidate B was a community psychologist.

Format

A list containing :

data matrix of size 5738x5 containing the 5738 observed full ranks in ranking representation. The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

For example, if the ranking representation of a rank is (4,3,1,2,5), it means that judge ranks the first object in 4th position, second object in 3rd position, ...

frequency matrix of size 120x6. Each row corresponds to one of the different observed rank. The first fifth columns contains the observed ranks (ordering representation) and the sixth column contains the frequency of observation.

m vector with the size of the ranks (5 here).

Source

"Group representations in probability and statistics", P. Diaconis, 1988.

Examples

```
data(APA)
```

---

big4 *rank data : big4*

Description

This dataset is composed of the rankings (in ranking notation) of the "Big Four" English football teams (A: Manchester, B: Liverpool, C: Arsenal, D: Chelsea) to the English Championship (Premier League) and according to the UEFA coefficients (statistics used in Europe for ranking and seeding teams in international competitions), from 1993 to 2013.

In 2000-2001, Arsenal and Chelsea had the same UEFA coefficient and then are tied. UEFA ranking is (1, 4, 2, 2) for 2000-2001, what means that Manchester United is the first, Liverpool is the last, and the two intermediate positions are for Arsenal and Chelsea in an unknown order.

In 2009-2010, Liverpool and Arsenal have also the same UEFA coefficient, the ranking is (1, 2, 2, 4).

Format

A list containing :

data A matrix of size 21*8 containing the 21 Premier League seasons. Each row corresponding to one ranking (ranking representation).

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

For example, if the ranking representation of a rank is (4,3,1,2,5), it means that judge ranks the first object in 4th position, second object in 3rd position, ...

frequency matrix of size 21*9. Each row corresponds to one of the 21 different observed rankings, and the last column contains the observation frequency.

m the size of the rankings (m=c(4,4)).

Source

http://en.wikipedia.org/wiki/Premier_League#.22Big_Four.22_dominance_.282000s.29

http://www.uefa.com/memberassociations/uefarankings/club/index.html

Examples

```
data(big4)
```

`convertRank` *change the representation of a rank*

Description

convertRank converts a rank from its ranking representation to its ordering representation, and vice-versa. The function does not work with partial ranking. The transformation to convert a rank from ordering to ranking representation is the same that from ranking to ordering representation, there is no need to precise the representation of rank x.

Usage

```
convertRank(x)
```

Arguments

x	a rank (vector) datum either in its ranking or ordering representation.

Details

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

The ordering representation o=(o_1,...,o_m) means that object o_i is in the ith position.

Let us consider the following example to illustrate both notations: a judge, which has to rank three holidays destinations according to its preferences, O1 = Countryside, O2 =Mountain and O3 = Sea, ranks first Sea, second Countryside, and last Mountain. The ordering result of the judge is o = (3, 1, 2) whereas the ranking result is r = (2, 3, 1).

Value

a rank (vector) in its ordering representation if its ranking representation has been given in input of convertRank, and vice-versa.

Author(s)

Julien Jacques

Examples

```
x <- c(2, 3, 1, 4, 5)
convertRank(x)
```

`criteria` *criteria estimation*

Description

This function estimates the loglikelihood of a mixture of multidimensional ISR model, as well as the BIC and ICL model selection criteria.

Usage

```
criteria(data, proportion, pi, mu, m, Ql = 500, Bl = 100, IC = 1,
  nb_cpus = 1)
```

Arguments

`data`	a matrix in which each row is a rank (partial or not; for partial rank, missing elements of a rank are put to 0).
`proportion`	a vector (which sums to 1) containing the K mixture proportions.
`pi`	a matrix of size K*p, where K is the number of clusters and p the number of dimension, containing the probabilities of a good comparaison of the model (dispersion parameters).
`mu`	a matrix of size K*sum(m), containing the modal ranks. Each row contains the modal rank for a cluster. In the case of multivariate ranks, the reference rank for each dimension are set successively on the same row.
`m`	a vector containing the size of ranks for each dimension.
`Ql`	number of iterations of the Gibbs sampler used for the estimation of the log-likelihood.
`Bl`	burn-in period of the Gibbs sampler.
`IC`	number of run of the computation of the loglikelihood.
`nb_cpus`	number of cpus for parallel computation

Value

a list containing:

`ll`	the estimated log-likelihood.
`bic`	the estimated BIC criterion.
`icl`	the estimated ICL criterion.

Author(s)

Quentin Grimonprez

Examples

```
data(big4)
res = rankclust(big4$data, m = big4$m, K = 2, Ql = 100, Bl = 50, maxTry = 2)
if(res@convergence)
crit = criteria(big4$data, res[2]@proportion, res[2]@pi, res[2]@mu,
                big4$m, Ql = 200, Bl = 100)
```

`distCayley` *Cayley distance between two ranks*

Description

The Cayley distance between two ranks x and y is the minimum number of transpositions required to transform the ranking x into y.

Usage

```
distCayley(x, y)
```

Arguments

`x, y`	two ranks of size m.

Value

the Cayley distance between x and y.

Author(s)

Julien Jacques

Examples

```
x <- 1:5
y <- c(2, 3, 1, 4, 5)
distCayley(x, y)
```

distHamming *Hamming distance between two ranks*

Description

The Hamming distance between two ranks x and y is the number of difference between the two ranks. For example, the Hamming's distance between x=(1,4,2,5,3) and y=(1,3,4,5,2) is 3 because, only 1 and 5 have the same place in both ranks.

Usage

```
distHamming(x, y)
```

Arguments

x, y	two ranks of size m.

Value

an integer, the Hamming distance between x and y.

Author(s)

Julien Jacques

Examples

```
x <- 1:5
y <- c(2, 3, 1, 4, 5)
distHamming(x, y)
```

distKendall *Kendall distance between two ranks*

Description

The Kendall distance between two ranks is the number of pairs that are in different order in the two ranks.

Usage

```
distKendall(x, y, type = "ordering")
```

Arguments

x, y	two ranks of size m.
type	type of the rank representation ("ordering" ou "ranking").

Value

an integer, the Kendall distance between x and y.

Author(s)

Julien Jacques

References

A New Measure of Rank Correlation, M. G. Kendall

Examples

```
x <- 1:5
y <- c(2, 3, 1, 4, 5)
distKendall(x, y, type = "ordering")
```

---

distSpearman *Spearman distance between two ranks*

Description

The Spearman distance is the square of Euclidean distance between two rank vector.

Usage

```
distSpearman(x, y)
```

Arguments

x, y	two ranks of size m.

Value

an integer, the Spearman distance between x and y.

Author(s)

Julien Jacques

Examples

```
x <- 1:5
y <- c(2, 3, 1, 4, 5)
distSpearman(x,y)
```

`eurovision` *Multidimensionnal partial rank data : eurovision*

Description

This dataset contains the ranking of the 8 common finalists of the Eurovision song contest from 2007 to 2012:

A: France, B:Germany, C:Greece, D:Romania, E:Russia, F:Spain, G:Ukraine, H:United Kingdom.

The number of rankings is 33, corresponding to the 33 European countries having participated to this six editions of the contest.

All the rankings are partial since none country has ranked this 8 countries in its 10 preferences. Missing ranking elements are zeros.

Format

A list containing:

data A matrix of size 34*48. Each row corresponds to the ranking representation of a multidimensionnal ranking. Columns 1 to 8 correspond to the 2007 contest, columns 9 to 18 to the 2008 contest, etc...

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

For example, if the ranking representation of a rank is (4,3,1,2,5), it means that judge ranks the first object in 4th position, second object in 3rd position, ...

frequency A matrix of size 34*49 containing the differents multidimensionnal rankings. The 48 first columns are the same as in data, and the last column contains the frequency (1 for all ranks).

m a vector with the sizes of ranks for each dimension.

Source

https://eurovision.tv

Examples

```
data(eurovision)
```

frequence *Convert data storage*

Description

This function takes in input a matrix containing all the observed ranks (a rank can be repeated) and returns a matrix containing all the different observed ranks with their observation frequencies (in the last column).

Usage

```
frequence(X, m = ncol(X))
```

Arguments

X	a matrix containing ranks.
m	a vector with the size of ranks of each dimension.

Value

A matrix containing each different observed ranks with its observation frequencies in the last column.

Author(s)

Quentin Grimonprez

Examples

```
X <- matrix(1:4, ncol = 4, nrow = 5, byrow = TRUE)
Y <- frequence(X)
Y
```

khi2 *khi2 test*

Description

This function computes the p-value of the khi2 adequation test (only for univariate data).

Usage

```
khi2(data, proportion, mu, pi, nBoot = 1000)
```

Arguments

data	a matrix in which each row is a rank of size m.
proportion	a vector (which sums to 1) containing the K mixture proportion.
mu	a matrix of size K*m, where m is the size of a rank, containing the modal rankings of the model (position parameters).
pi	a vector of size K, where K is the number of clusters, containing the probabilities of a good paired comparaison of the model (dispersion parameters).
nBoot	number of bootstrap iterations used to estimate the khi2 adequation test p-value.

Value

a real, the p-value of the khi2 adequation test.

Author(s)

Quentin Grimonprez

Examples

```
proportion  <- c(0.4, 0.6)
pi <- c(0.8, 0.75)
mu <- matrix(c(1, 2, 3, 4, 4, 2, 1, 3), nrow = 2, byrow = TRUE)
# simulate a data set with declared parameters.
data <- rbind(simulISR(proportion[1] * 100, pi[1], mu[1,]),
simulISR(proportion[2] * 100, pi[2], mu[2,]))
pval <- khi2(data, proportion, mu, pi)
```

---

kullback *Kullback-Leibler divergence*

Description

This function computes the Kullback-Leibler divergence between two mixtures of multidimensional ISR distributions.

Usage

```
kullback(proportion1, pi1, mu1, proportion2, pi2, mu2, m)
```

Arguments

proportion1, proportion2	vectors (which sums to 1) containing the K mixture proportions.
pi1, pi2	matrices of size K*p, where K is the number of clusters and p the number of dimension, containing the probabilities of a good comparaison of the model (dispersion parameters).

mu1, mu2	matrices of size K*sum(m), containing the modal ranks. Each row contains the modal rank for a cluster. In the case of multivariate ranks, the reference rank for each dimension are set successively on the same row.
m	a vector containing the size of ranks for each dimension.

Value

a real, the Kullback-Leibler divergence.

Author(s)

Quentin Grimonprez

References

http://en.wikipedia.org/wiki/Kullback

Examples

```
proportion1 <- c(0.4, 0.6)
pi1 <- matrix(c(0.8, 0.75), nrow = 2)
mu1 <- matrix(c(1, 2, 3, 4, 4, 2, 1, 3), nrow = 2, byrow = TRUE)
proportion2 <- c(0.43, 0.57)
pi2 <- matrix(c(0.82, 0.7), nrow = 2)
mu2 <- matrix(c(1, 2, 3, 4, 4, 2, 1, 3), nrow = 2, byrow = TRUE)
dK <- kullback(proportion1, pi1, mu1, proportion2, pi2, mu2, 4)
```

Output-class *Constructor of Output class*

Description

This class contains a result of a run. Let K be the total number of cluster, p the number of dimension m the p-vector containing the size of each dimension.

Details

proportion a K-vector of proportions.

pi a K*p-matrix composed of the scale parameters.

mu a matrix with K lines and sum(m) columns in which line k is composed of the location parameters of cluster k.

ll the estimated log-likelihood.

bic the estimated BIC criterion.

icl the estimated ICL criterion.

tik a n*K-matrix containing the estimation of the conditional probabilities for the observed ranks to belong to each cluster.

partition a n-vector containing the partition estimation resulting from the clustering.

entropy a n*2-matrix containing for each observation its estimated cluster and its entropy. The entropy output illustrates the confidence in the clustering of each observation (a high entropy means a low confidence in the clustering)..

probability a n*2-matrix similar to the entropy output, containing for each observation its estimated cluster and its probability p(xi; mk, pk) given its cluster. This probability is estimated using the last simulation of the presentation orders used for the likelihood approximation. The probability output exhibits the best representative of each cluster.

convergence a boolean indicating if none problem of empty class has been encountered.

partial a boolean indicating the presence of partial rankings or ties.

partialRank a matrix containing the full rankings, estimated using the within cluster ISR parameters when the ranking is partial. When ranking is full, partialRank simply contains the observed ranking. Available only in presence of at least one partial ranking.

distanceProp Distances (MSE) between the final estimation and the current value at each iteration of the SEM-Gibbs algorithm (except the burning phase) for proportions. A list of Qsem-Bsem elements, each element being a K*p-matrix.

distancePi Distances (MSE) between the final estimation and the current value at each iteration of the SEM-Gibbs algorithm (except the burning phase) for scale parameters. A list of Qsem-Bsem elements, each element being a K*p-matrix.

distanceMu Distances (Kendall distance) between the final estimation and the current value at each iteration of the SEM-Gibbs algorithm (except the burning phase) for proportions. A list of Qsem-Bsem elements, each element being a K*p-matrix.

distanceZ a vector of size Qsem-Bsem containing the rand index between the final estimated partition and the current value at each iteration of the SEM-Gibbs algorithm (except the burning phase). Let precise that the rand index is not affected by label switching.

distancePartialRank Kendall distance between the final estimation of the partial rankings (missing positions in such rankings are estimated) and the current value at each iteration of the SEM-Gibbs algorithm (except the burning phase). distancePartialRank is a list of Qsem-Bsem elements, each element being a matrix of size n*p. Available only in presence of at least one partial ranking.

proportionInitial a vector containing the initialization of proportions in the algorithm.

piInitial a matrix containing the initialization of the probabilities of good paired comparison in the algorithm.

muInitial a matrix containing the initialization of modal rankings in the algorithm.

partialRankInitial a matrix containing the initialization of the partial rankings in the algorithm.

`probability` *Probability computation*

Description

This function computes the probability of x according to a multivariate ISR o parameter mu and pi.

Usage

```
probability(x, mu, pi, m = length(x))
```

Arguments

x	a vector or a matrix of 1 row containing the rankings in ranking notation (see Details or convertRank function). The rankings of each dimension are placed end to end. x must contain only full ranking (no partial or tied).
mu	a vector of length sum(m) or a matrix of size 1*sum(m), containing the modal ranks in ranking notation (see Details or convertRank function). The rankings of each dimension are placed end to end. mu must contain only full ranking (no partial or tied).
pi	a vector of size p, where p is the number of dimension, containing the probabilities of a good comparaison of the model (dispersion parameters).
m	a vector containing the size of ranks for each dimension.

Details

The ranks have to be given to the package in the ranking notation (see convertRank function), with the following convention :

- missing positions are replaced by 0

- tied are replaced by the lowest position they share"

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

The ordering representation o=(o_1,...,o_m) means that object o_i is in the ith position.

Let us consider the following example to illustrate both notations: a judge, which has to rank three holidays destinations according to its preferences, O1 = Countryside, O2 =Mountain and O3 = Sea, ranks first Sea, second Countryside, and last Mountain. The ordering result of the judge is o = (3, 1, 2) whereas the ranking result is r = (2, 3, 1).

Value

the probability of x according to a multivariate ISR o parameter mu and pi.

Author(s)

Quentin Grimonprez

Examples

```
m <- c(4, 5)
x = mu <- matrix(nrow = 1, ncol = 9)
x[1:4] = c(1, 4, 2, 3)
x[5:9] = c(3, 5, 2, 4, 1)
mu[1:4] = 1:4
mu[5:9] = c(3, 5, 4, 2, 1)
pi <- c(0.75, 0.82)
```

```
prob <- probability(x, mu, pi, m)
prob
```

---

quiz *Multidimensionnal rank data : quiz*

Description

This dataset contains the answers of 70 students (40 of third year and 30 of fourth year) from Polytech'Lille (statistics engineering school, France) to the four following quizzes:

Literature Quiz This quiz consists of ranking four french writers according to chronological order: A=Victor Hugo, B=Moliere, C=Albert Camus, D=Jean-Jacques Rousseau.

Football Quiz This quiz consists of ranking four national football teams according to increasing number of wins in the football World Cup: A=France, B=Germany, C=Brazil, D=Italy.

Mathematics Quiz This quiz consists of ranking four numbers according to increasing order: A=pi/3, B=log(1), C=exp(2), D=(1+sqrt(5))/2.

Cinema Quiz This quiz consists of ranking four Tarentino's movies according to chronological order: A=Inglourious Basterds, B=Pulp Fiction, C=Reservoir Dogs, D=Jackie Brown.

Format

A list containing:

data a matrix of size 70*16. The student's answers are in row and the 16 columns correspond to the 4 rankings (for the 4 quizzes) of size 4 (ranking representation).

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

For example, if the ranking representation of a rank is (4,3,1,2,5), it means that judge ranks the first object in 4th position, second object in 3rd position, ...

frequency a matrix of size 63*17. Each row corresponds to one of the 63 differents observed rankings (ranking representation). Each row contains 4 ranks of size 4 and a last column for the frequency.

m a vector with the sizes of the ranks for each dimension.

Source

Julien Jacques

Examples

```
data(quiz)
```

rankclust *model-based clustering for multivariate partial ranking*

Description

This functions estimates a clustering of ranking data, potentially multivariate, partial and containing tied, based on a mixture of multivariate ISR model [2]. By specifying only one cluster, the function performs a modelling of the ranking data using the multivariate ISR model. The estimation is performed thanks to a SEM-Gibbs algorithm in the general case.

Usage

```
rankclust(data, m = ncol(data), K = 1, criterion = "bic",
  Qsem = 100, Bsem = 20, RjSE = m * (m - 1)/2, RjM = m * (m - 1)/2,
  Ql = 500, Bl = 100, maxTry = 3, run = 1, detail = FALSE)
```

Arguments

data	a matrix in which each row is a ranking (partial or not; for partial ranking, missing elements must be 0 or NA. Tied are replaced by the lowest position they share). For multivariate rankings, the rankings of each dimension are placed end to end in each row. The data must be in ranking notation (see Details or convertRank functions).
m	a vector composed of the sizes of the rankings of each dimension (default value is the number of column of the matrix data).
K	an integer or a vector of integer with the number of clusters.
criterion	criterion "bic" or "icl", criterion to minimize for selecting the number of clusters.
Qsem	the total number of iterations for the SEM algorithm (defaut value=40).
Bsem	burn-in period for SEM algorithm (default value=10).
RjSE	a vector containing, for each dimension, the number of iterations of the Gibbs sampler used both in the SE step for partial rankings and for the presentation orders generation (default value=mj(mj-1)/2).
RjM	a vector containing, for each dimension, the number of iterations of the Gibbs sampler used in the M step (default value=mj(mj-1)/2)
Ql	number of iterations of the Gibbs sampler for estimation of log-likelihood (default value=100).
Bl	burn-in period for estimation of log-likelihood (default value=50).
maxTry	maximum number of restarts of the SEM-Gibbs algorithm in the case of non convergence (default value=3).
run	number of runs of the algorithm for each value of K.
detail	boolean, if TRUE, time and others informations will be print during the process (default value FALSE).

Details

The ranks have to be given to the package in the ranking notation (see convertRank function), with the following convention :

- missing positions are replaced by 0

- tied are replaced by the lowest position they share"

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

The ordering representation o=(o_1,...,o_m) means that object o_i is in the ith position.

Let us consider the following example to illustrate both notations: a judge, which has to rank three holidays destinations according to its preferences, O1 = Countryside, O2 =Mountain and O3 = Sea, ranks first Sea, second Countryside, and last Mountain. The ordering result of the judge is o = (3, 1, 2) whereas the ranking result is r = (2, 3, 1).

Value

An object of class rankclust (See `Output-class` and `Rankclust-class`). If the output object is named `res`. You can access the result by res[number of groups]@slotName where `slotName` is an element of the class Output.

Author(s)

Quentin Grimonprez

References

[1] C.Biernacki and J.Jacques (2013), A generative model for rank data based on sorting algorithm, Computational Statistics and Data Analysis, 58, 162-176.

[2] J.Jacques and C.Biernacki (2012), Model-based clustering for multivariate partial ranking data, Inria Research Report n 8113.

See Also

See `Output-class` and `Rankclust-class` for available output.

Examples

```
data(big4)
result = rankclust(big4$data, K = 2, m = big4$m, Ql = 200, Bl = 100, maxTry = 2)
```

Rankclust-class *Constructor of Rankclust class*

Description

This class contains results of rankclust function.

Details

K a vector of the number of clusters.

data the data used for clustering.

criterion the model selection criterion used.

convergence a boolean indicating if none problem of empty class has been encountered (for any number of clusters).

results a list of Output-class, containing the results for each number of clusters (one element of the list is associated to one number of clusters).

If res is the result of rankclust(), each slot of results can be reached by res[k]@slotname, where k is the number of clusters and slotname is the name of the slot we want to reach (see Output-class). For the slots ll, bic, icl, res["slotname"] returns a vector of size K containing the values of the slot for each number of clusters.

show *show function.*

Description

This function shows the elements of a given object.

Usage

```
## S4 method for signature 'Output'
show(object)

## S4 method for signature 'Rankclust'
show(object)
```

Arguments

object	an object of class Output or Rankclust.

`simulISR` *simulate a sample of ISR(pi,mu)*

Description

This function simulates univariate rankings data (ordering representation) according to the ISR(pi,mu).

Usage

```
simulISR(n, pi, mu)
```

Arguments

`n`	size of the sample.
`pi`	dispersion parameter: probability of correct paired comparaison according to mu.
`mu`	position parameter: modal ranking in ordering representation.

Details

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

The ordering representation o=(o_1,...,o_m) means that object o_i is in the ith position.

Let us consider the following example to illustrate both notations: a judge, which has to rank three holidays destinations according to its preferences, O1 = Countryside, O2 =Mountain and O3 = Sea, ranks first Sea, second Countryside, and last Mountain. The ordering result of the judge is o = (3, 1, 2) whereas the ranking result is r = (2, 3, 1).

You can see the convertRank function to convert the simualted ranking drom ordering to ranking representation.

Value

a matrix with simulated ranks.

Author(s)

Julien Jacques

References

[1] C.Biernacki and J.Jacques (2013), A generative model for rank data based on sorting algorithm, Computational Statistics and Data Analysis, 58, 162-176.

Examples

```
x <- simulISR(30, 0.8, 1:4)
```

sports *rank data : sports*

Description

This data set is due to Louis Roussos who asked 130 students at the University of Illinois to rank seven sports according to their preference in participating: A = Baseball, B = Football, C = Basketball, D = Tennis, E = Cycling, F = Swimming, G = Jogging.

Format

A list containing :

data a matrix containing 130 ranks of size 7 in ranking representation.
The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.
For example, if the ranking representation of a rank is (4,3,1,2,5), it means that judge ranks the first object in 4th position, second object in 3rd position, ...

frequency a matrix with 123 differents ranks of size 7. In each row the first 7 columns correspond to one observed ranking and the last column contains the observation frequency.

m the size of the rankings (m=7).

Source

J.I. Marden. "Analyzing and modeling rank data, volume 64 of Monographs on Statistics and Applied Probability". Chapman & Hall, London, 1995.

Examples

```
data(sports)
```

summary *summary function.*

Description

This function This function gives the summary of output from `rankclust`.

Usage

```
## S4 method for signature 'Rankclust'
summary(object, ...)
```

Arguments

`object`	output object from `rankclust`.
`...`	Not used.

unfrequence *Convert data*

Description

This function takes in input a matrix in which the m first columns are the different observed ranks and the last column contains the observation frequency, and returns a matrix containing all the ranks (ranks with frequency>1 are repeated).

Usage

```
unfrequence(data)
```

Arguments

data	a matrix containing rankings and observation frequency.

Value

a matrix containing all the rankings.

Examples

```
data(quiz)
Y <- unfrequence(quiz$frequency)
Y
```

words *rank data : words*

Description

The data was collected under the auspices of the Graduate Record Examination Board. A sample of 98 college students were asked to rank five words according to strength of association (least to most associated) with the target word "Idea": A = Thought, B = Play, C = Theory, D = Dream and E = Attention.

Format

A list containing :

data A matrix of size 98*5 containing the 98 answers. Each row corresponding to one ranking (ranking representation).

The ranking representation r=(r_1,...,r_m) contains the ranks assigned to the objects, and means that the ith object is in r_ith position.

For example, if the ranking representation of a rank is (4,3,1,2,5), it means that judge ranks the first object in 4th position, second object in 3rd position, ...

frequency matrix of size 15*6. Each row corresponds to one of the 15 different observed rankings, and the last column contains the observation frequency.

m the size of the rankings (m=5).

Source

M.A. Fligner and J.S. Verducci. "Distance based ranking models". J. Roy. Statist. Soc. Ser. B, 48(3):359-369, 1986.

Examples

```
data(sports)
```

[*Getter method for rankclust output*

Description

This is overloading of square braces to extract values of various slots of the output from the function `rankclust`.

Usage

```
## S4 method for signature 'Rankclust'
x[i]
```

Arguments

`x`	object from which to extract element(s) or in which to replace element(s).
`i`	the number of cluster of the element we want to extract.

Index