Package ‘StatMethRank’

February 19, 2015

Type Package
Title Statistical Methods for Ranking Data
Version 1.0
Date 2014-05-27
Author Li Qinglong
Maintainer Li Qinglong <liqinglong@163.com>
Description Functions and datasets to support Statistical Methods for Ranking Data by Mayer Alvo and Philip L.H. Yu(2014)
Depends MASS, utils, rjags, pmr
License GPL-2
NeedsCompilation no
Repository CRAN
Date/Publication 2014-07-10 08:45:39

R topics documented:

StatMethRank-package .................................................. 2
agreement.test .......................................................... 3
APA ................................................................. 3
big4year ............................................................. 4
case2freq .......................................................... 4
compatible.rankings .............................................. 5
Croon .............................................................. 6
freq2case .......................................................... 6
generate.combs .................................................. 7
generate.perms .................................................. 8
HKPOS ............................................................ 9
incomplete.rankings ........................................... 9
independence.test ............................................... 10
interaction.test ............................................... 11
is.compatible .................................................. 12
msci ............................................................ 13
Description

Statistical Methods for Ranking Data

Details

Package:          StatMethRank
Type:             Package
Version:          1.0
Date:             2014-05-28
License:          GPL-2

Functions and datasets to support Alvo and Yu (2014), 'Statistical Methods for Ranking Data'

Author(s)

Li Qinglong
Maintainer: Li Qinglong <liqinglong0830@163.com>

References


See Also

pmr (http://cran.r-project.org/web/packages/pmr/index.html)
agreement.test

Description
This function performs a test of agreement among groups.

Usage
agreement.test(data, method = c("spearman", "kendall"))

Arguments
- data: a data frame of the frequencies of all possible rankings given by different groups
- method: whether the test is based on Spearman metric or Kendall metric

Value
a list of test statistics

Author(s)
Li Qinglong <liqinglong0830@163.com>

References
Intergroup Diversity and Concordance for Ranking Data: An Approach via Metrics for Permuta-
tions, Paul D. Feigin and Mayer Alvo

Examples
data(Sutton)
agreement.test(Sutton, method = "spearman")
agreement.test(Sutton, method = "kendall")

APA
American Psychological Association Election Data

Description
In 1980, the American Psychological Association conducted an election at which five candidates (A, B, C, D and E) were running for president and voters were asked to rank order all of the candidates. Among those voters, 5738 gave complete rankings. These complete rankings are considered here. The data are given in Diaconis (1988). Suppose that lower rank implies more favorable.
case2freq

Format
A data frame with 120 rows and 6 variables

Details
• freq. The frequencies of each distinguish ranking.
• A, B, C, D and E. The ranks of the five candidates.

big4year

Data on the big four EPL Teams

Description
The English Premier League (EPL) is a famous professional league for association soccer clubs in the UK. The so-called "Big Four" soccer clubs are Arsenal, Chelsea, Liverpool, and Manchester United, have dominated the top four spots since the 1996-1997 season. Wikipedia documented the results of the "Big Four" since the start of the Premier League in the 1992-93 season. The rankings of these four English Premier League teams from the 1992-93 season to the 2012-13 season are listed in this data frame. The first row of the data means that there is one season that Arsenal ranked at the top of EPL, Chelsea the second, Manchester United the third and Liverpool the fourth. We may test the hypothesis that the rankings observed are random. On the other hand, through the use of modeling we may try to determine how the rankings cluster.

Format
A data frame with 21 rows and 4 variables

Details
• Arsenal, Chelsea, Liverpool, and Manchester United. The ranks of the "Big Four" clubs

caseRfreq

Convert raw ranking data(case form) to ranking data with rank frequencies

Description
Convert raw ranking data(case form) to ranking data with rank frequencies

Usage
case2freq(x)

Arguments
x A data frame or a matrix(case form), each row is a rank.
compatible.rankings

Value
A data frame, each row contains a rank and the corresponding frequency.

Author(s)
Li Qinglong <liqinglong0830@163.com>

Examples
```r
data(APA)
cases = freq2case(APA, freq.col = 1)
freqs = case2freq(cases)
```

compatible.rankings  Set of compatible rankings

Description
This function generates the set of complete permutations compatible with the input_ranking.

Usage
```r
compatible.rankings(input_ranking)
```

Arguments
- `input_ranking`: a vector, using NA to stand for the missing ranks

Value
a matrix or a vector, each column is a complete ranking.

Author(s)
Li Qinglong <liqinglong0830@163.com>

Examples
```r
u_star = c(2, NA, 3, 4, 1)
C_set = compatible.rankings(u_star)
is.compatible(C_set, u_star)
```
Description

M.A. Croon conducted a survey in Germany on the political goals on the Government. The data set contains 2262 rankings, in the aggregated form, of four goals.

Format

A matrix with 24 rows and 5 columns

Details

- 1, Maintain order in nation
- 2, Give people more say in Government decisions
- 3, Fight rising prices
- 4, Protect freedom of speech

Description

Convert frequency table to raw data(case form). This is a useful function to convert the data written by Michael Friendly.

Usage

freq2case(x, var.names = NULL, freq.name = "Freq", freq.col = NULL, ...)

Arguments

- x: A table object, or a data frame in frequency form containing ranks and one numeric variable representing the frequency for that rank.
- var.names: A list of variable names for the factors, if you wish to override those already in the table.
- freq.name: The name of the frequency variable in the table.
- freq.col: The column index of the frequencies.
- ...: Other arguments passed down to type.convert.
**generate.combs**

**Value**

A data frame containing the factors in the table and as many observations as are represented by the total of the freq variable.

**Author(s)**

Michael Friendly, edited by Li Qinglong <liqinglong0830@163.com>

**References**


**Examples**

```r
data(APA)
cases = freq2case(APA, freq.col = 1)
freqs = case2freq(cases)
```

**generate.combs**  
Generate all possible combinations of k elements out of n

**Description**

This function generates all combinations of n elements taken k at a time. It is just a reference to the combn function in utils package.

**Usage**

`generate.combs(n, k = 1)`

**Arguments**

- `n` number of the whole elements
- `k` number of elements to choose (default 1)

**Value**

a matrix or a vector

**Author(s)**

Scott Chasalow wrote the original in 1994 for S; R package combinat and documentation by Vince Carey stvjc@channing.harvard.edu; small changes by the R core team.

**See Also**

utils::combn
generate.perms

**Examples**

generate.combs(10, 6)

---

generate.perms  Generate all possible permutations of k elements out of n

**Description**

This function generates all permutations of n elements taken k at a time. An efficient way to generate n! permutations of vec in a Lexicographic order without recursive algorithm.

**Usage**

generate.perms(n, k = n, vec = 1:n)

**Arguments**

- **n**: number of the whole elements
- **k**: number of elements to permute (default the same as n)
- **vec**: the source vector of length n (default as c(1:n))

**Value**

A matrix or a vector, each row is a permutation

**Author(s)**

Li Qinglong <liqinglong0830@163.com>

**See Also**

generate.combn

**Examples**

generate.perms(10, 6)
In a public opinion survey held in 1999 in Hong Kong, it was of interest to determine whether the education level of the respondents is related to the level of dissatisfaction of the Policy Address of the Chief Executive of the Hong Kong Special Administrative Region.

A matrix (contingency table) of 6-by-6

- **Edu_Level.** Education Level of the respondents. 1: Primary, 2: Lower Secondary, 3: Upper Secondary, 4: Post-secondary (non-degree), 5: Post-secondary (degree), NA: missing data.
- **Response.** Level of dissatisfaction of the Policy Address of the Chief Executive of the Hong Kong Special Administrative Region. The response is an ordinal variable having seven options as follows: 1: very satisfied, 2: satisfied, 3: neutral, 4: unsatisfied, 5: very unsatisfied, 6: not sure and 7: refuse to answer. Options 6 and 7 were combined and listed as NA.

This function generates all the possible incomplete rankings of k elements out of n complete elements. E.g. c(1, 2, NA) is an incomplete ranking out with 2 non-missing elements of 3 elements

**Usage**

incomplete.rankings(n, k)

**Arguments**

- n number of complete elements
- k number of non-missing elements in the incomplete ranking

**Value**

a matrix each of whose column is a possible incomplete ranking
**Author(s)**

Li Qinglong <liqinglong0830@163.com>

**Examples**

```r
incomplete.rankings(5, 3)
```

---

**Description**

This function performs a nonparametric test of ranking data based on the correlation. This function can be applied to the ranking data with missing ranks and tie ranks.

**Usage**

```r
independence.test(X1, X2, method = c("spearman", "kendall"))
```

**Arguments**

- `X1`: a vector, using NA to stand for the missing ranks
- `X2`: the same as `X1`
- `method`: whether the test is based on Spearman correlation or Kendall correlation

**Value**

a list of the test statistics

**Author(s)**

Li Qinglong <liqinglong0830@163.com>

**References**

Rank Correlation Methods for Missing Data, Mayer Alvo and Paul Cablio
Nonparametric Rank Tests for Independence in Opinion Surveys, Philip L.H. Yu, K.F. Lam, and Mayer Alvo

**Examples**

```r
Arith = c(14, 18, 23, 26, 27, 30, 40, NA, NA)
Lang = c(28, 14, 46, NA, 53, NA, 54, 50, NA)
independence.test(Arith, Lang, method = "spearman")
independence.test(Arith, Lang, method = "kendall")
```
**Description**

This function performs a test of interaction of ranking data.

**Usage**

`interaction.test(X)`

**Arguments**

`X`  
a I * J * N array, two-factor design dataset. X[i, j, n] denotes the response of the n-th replicate in the (i, j) cell.

**Value**

a list of test statistic and the p value

**Author(s)**

Li Qinglong <liqinglong0830@163.com>

**References**

A Nonparametric test for interaction in two-way layouts, Xin Gao and Mayer Alvo

**Examples**

```r
# Box-cox data revisited
boxcoxdat = array(0, c(3, 4, 4))
boxcoxdat[1, , ] = matrix(c(0.31, 0.82, 0.43, 0.45,
                           0.45, 1.10, 0.45, 0.71,
                           0.46, 0.88, 0.63, 0.66,
                           0.43, 0.72, 0.76, 0.62), nrow = 4)
boxcoxdat[2, , ] = matrix(c(0.36, 0.92, 0.44, 0.56,
                           0.29, 0.61, 0.35, 1.02,
                           0.40, 0.49, 0.31, 0.71,
                           0.23, 1.24, 0.40, 0.38), nrow = 4)
boxcoxdat[3, , ] = matrix(c(0.22, 0.30, 0.23, 0.30,
                           0.21, 0.37, 0.25, 0.36,
```

is.compatible

0.18, 0.38, 0.24, 0.31,
0.23, 0.29, 0.22, 0.33, nrow = 4
interaction.test(boxcoxdat)

is.compatible  Is compatible with the candidate_ranking

Description
This function judges whether the complete rankings are compatible with an incomplete ranking.

Usage
is.compatible(complete_ranking, candidate_ranking)

Arguments
complete_ranking
a matrix or a vector, each column is a complete ranking.
candidate_ranking
a vector, using NA to stand for the missing ranks.

Value
a vector of TRUEs and FALSEs

Author(s)
Li Qinglong <liqinglong0830@163.com>

Examples
u_star = c(2, NA, 3, 4, 1)
C_set = compatible.rankings(u_star)
is.compatible(C_set, u_star)
**msci**

*Mean, Standard deviation, Confident Intervals*

---

**Description**

Get the Mean, Standard deviation and Confidence Intervals of a sampling trace.

**Usage**

```r
msci(df, sig = 0.05)
```

**Arguments**

- `df`: a data frame, each column is a sampling trace.
- `sig`: significance level, default is 0.05.

**Value**

A data frame with MSCI of each trace.

**Author(s)**

Li Qinglong <liqinglong0830@163.com>

---

**mvnos.model**

*Multivariate Normal Order-statistics Model.*

---

**Description**

Using MCMC methods to fit the MVNOS model. Please install JAGS 3.X (http://mcmc-jags.sourceforge.net) and rjags (http://cran.r-project.org/web/packages/rjags/index.html) at first.

**Usage**

```r
mvnos.model(y, p, Z, beta0 = NULL, A0 = NULL, alpha = NULL, P = NULL, BURN_IN_ITERATIONS = 1000, MAX_ITERATIONS = 10000, DRAW_CYCLE = 20)
```
Arguments

- **y**: an n*k matrix, observed data, each row is an individual’s rank of items
- **p**: number of parameters in MVNOS model
- **Z**: a n*k*p array of covariates associated with all judges
- **beta0**: a 1*p matrix, prior normal distribution mean parameters
- **A0**: a p*p matrix, prior normal distribution variance-covariance matrix
- **alpha**: scalar, prior Wishart distribution degree of freedom
- **P**: a (k-1)*(k-1) matrix, prior Wishart distribution scale matrix

**BURN_IN_ITERATIONS**: number of iterations to burn-in at first

**MAX_ITERATIONS**: full sample iterations

**DRAW_CYCLE**: reduce the full sample by draw-cycle (e.g. draw every 20th draw from the full sample)

Value

A list of Gibbs sampling traces

Author(s)

Li Qinglong <liqinglong0830@163.com>

References


Examples

```r
# APA data application
# It will take about 10 minutes to run the demo.
data(APA)
y = freq2case(APA, freq.col = 1)
y = 6 - y  # number of observed judges
n = dim(y)[1]
# number of items
k = dim(y)[2]
# number of parameters of beta
p = k
beta0 = rep(0, p)
alpha = k + 1
A0 = diag(100, ncol = p, nrow = p)
P = diag(k + 1, ncol = k - 1, nrow = k - 1)
# Construct Z
Z = array(0, dim = c(n, k, p))
for (j in 1:n) {
  
```
mwdbm

Fit a mixture weighted distance-based model

Description

This function computes fitting of mixture weighted distance-based model for the given data set of complete rankings.

Usage

mwdbm(dset, G, dset.agg = TRUE, dtype = "Kendall", noise = FALSE, iter = 100)

Arguments

dset data set of complete rankings
G number of clusters
dset.agg whether the data set is in the aggregated form (default as FALSE)
dtype type of the weighted distance measure (default as weighted Kendall’s)
noise whether a noise cluster is contained (default as FALSE)
iter number of iterations of the EM algorithm (default as 100)

Value

a list of the fitting result, containing the following objects: $clusterNum number of clusters (excluding the noise) $dtype type of the distance measure $noise whether a noise cluster is contained $iterNum actual number of iterations of the EM algorithm $convergence whether the complete-data loglikelihood converges $clusterProb probability of each cluster $modalRank modal rankings $weight weight vectors for clusters $trueLoglik the true loglikelihood by the fitted model $squaredPearsonStat the sum of squares of Pearson residuals

Author(s)

Yumin Zhang <zymneo@gmail.com>
Examples

data(Croon)
# Time comu
# mwdom(Croon, 3)

repmat

Replicate and tile array

Description

Just the same usage as repmat function in Matlab. This kind of functions can be easily found on the web.

Usage

repmat(A, M, N)

Arguments

A Matrix or vector to repeat.
M Number of row repetitions.
N Number of column repetitions.

Value

a matrix of M-by-N tiling of A.

Examples

repmat(c(1, 2), 6, 8)

Scores

Test scores in Language and Arithmetic for a group of 9 students

Description


Format

A data frame of 2 variables

Details

- Arith. Arithmetic scores of 9 students.
**Song**

<table>
<thead>
<tr>
<th>Song</th>
<th><strong>Song Data</strong></th>
</tr>
</thead>
</table>

**Description**

The "song" data set from Critchlow et al. (1991). Ninety-eight students were asked to rank 5 words, (1) score, (2) instrument, (3) solo, (4) benediction and (5) suit, according to the association with the word "song". However, the available data given in Critchlow et al. (1991) is in grouped format and the ranking of 15 students are unknown and hence discarded, resulting in 83 rankings.

**Format**

A data frame with 13 rows and 6 columns

**Details**

- freq. The frequencies of each student’s ordering.
- order1~order5 The ordering of the five words. (1) score, (2) instrument, (3) solo, (4) benediction and (5) suit

---

**Sutton**

<table>
<thead>
<tr>
<th>Sutton</th>
<th><strong>Sutton data on leisure preferences</strong></th>
</tr>
</thead>
</table>

**Description**

In a survey conducted in Florida, C.Sutton asked a group of female elderly for 14 white and 13 black retired women aged 70-79 "with which sex do you prefer to spend your leisure?" Each elderly ranked the three choices: A: male(s), B: female(s), C: both sexes, assigning rank 1 to the most desired choice, rank 2 to the next most desired rank 3 to the least desired choice. The first object in the ranking corresponds to "male", the second to "female" and the third to "both". It was desired to compare these two groups on leisure preferences and attitudes on retirement of the elderly.

**Format**

A data frame with 6 rows and 5 variables

**Details**

- Male, Female and Both. The ranks of the 3 choices
- White and Black. The frequencies of each ranking of the group
Compute the weighted distances between two data sets of rankings

**Description**

This function computes the weighted distances between two data sets of complete rankings. The results are put in the matrix form. The data set could be aggregated or not.

**Usage**

```r
wdmat(dset1, dset2 = dset1, dset1.agg = FALSE, dset2.agg = FALSE, 
      dtype = "Kendall", weight = NULL, modal = NULL)
```

**Arguments**

- `dset1`: one data set with each row being a ranking
- `dset2`: the other data set (default as `dset1`)
- `dset1.agg`: whether the data set is in the aggregated form (default as `FALSE`)
- `dset2.agg`: whether the data set is in the aggregated form (default as `FALSE`)
- `dtype`: type of the weighted distance measure (default as weighted Kendall’s)
- `weight`: weight vector (default as all components being 1)
- `modal`: the modal ranking (default as `c(1:k)`, `k` being the number of ranked items)

**Value**

A list whose first object is a vector about the aggregation status of the two data sets `c(dset1.agg, dset2.agg)`, and second object is a matrix of distances.

**Author(s)**

Yumin Zhang <zymneo@gmail.com>

**Examples**

```r
data(Croon)
wdmat(Croon, dset1.agg=TRUE, dset2.agg=TRUE)
```
Index

*Topic datasets
APA, 3
big4year, 4
Croon, 6
HKPOS, 9
Scores, 16
Song, 17
Sutton, 17

*Topic ranking
StatMethRank-package, 2

agreement.test, 3
APA, 3

case2freq, 4
compatible.rankings, 5
Croon, 6

dfreq2case, 6

generate.combs, 7
generate.perms, 8

HKPOS, 9

incomplete.rankings, 9
independence.test, 10
interaction.test, 11
is.compatible, 12

msci, 13
mvnos.model, 13
mwdbm, 15

repmat, 16

Scores, 16
Song, 17
StatMethRank (StatMethRank-package), 2