

Package ‘R2ucare’

October 12, 2022

License GPL (>= 2)

Title Goodness-of-Fit Tests for Capture-Recapture Models

Description Performs goodness-of-fit tests for capture-recapture models as described by Gimenez et al. (2018) <[doi:10.1111/2041-210X.13014](https://doi.org/10.1111/2041-210X.13014)>. Also contains several functions to process capture-recapture data.

Version 1.0.2

URL <https://github.com/oliviergimenez/R2ucare>

Depends R (>= 3.3.0)

Suggests knitr, rmarkdown, testthat

Imports stringr, RMark, stats, utils

Encoding UTF-8

RoxygenNote 7.2.0

VignetteBuilder knitr

BugReports <https://github.com/oliviergimenez/R2ucare/issues>

NeedsCompilation no

Author Olivier Gimenez [aut, cre] (<<https://orcid.org/0000-0001-7001-5142>>),
Jean-Dominique Lebreton [ctb],
Remi Choquet [ctb],
Roger Pradel [ctb]

Maintainer Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>

Repository CRAN

Date/Publication 2022-07-11 08:10:12 UTC

R topics documented:

coef_mixture	2
deviance_mixture	3
expval_table	4
gof_test	4
group_data	5

group_data_gen	6
ind_test_22	6
ind_test_rc	7
inv_logit_gen	8
marray	8
multimarray	9
overall_CJS	10
overall_JMV	11
pool2K	12
pooling_ct	13
pooling_mixtures	14
read_headed	14
read_inp	15
reconstitution	16
repmat	17
test2cl	17
test2ct	18
test3Gsm	19
test3Gsr	20
test3Gwbwa	21
test3sm	22
test3sr	23
testMitec	24
testMltec	25
ungroup_data	26
Index	27

coef_mixtures	<i>Estimation of multinomial mixture distributions parameters</i>
---------------	---

Description

This function performs maximum likelihood inference for multinomial mixture distributions.

Usage

```
coef_mixtures(Mp, Np)
```

Arguments

Mp	a matrix of mixtures (a row matrix if a vector)
Np	a matrix of bases (a row matrix if a vector)

Value

This function returns a list of maximum likelihood estimates for the cells of a mixture distribution:

P matrix of cell probabilities estimates for mixtures

PI matrix of mixture probabilities

GAM matrix of cell probabilities estimates for bases

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

References

Yantis, S., Meyer, D. E., and Smith, J. E. K. (1991). Analyses of multinomial mixture distributions: New tests for stochastic models of cognition and action. *Psychological Bulletin* 110, 350–374.

deviance_mixture	<i>Deviance of multinomial mixture distributions</i>
------------------	--

Description

This function calculates the deviance of multinomial mixture distributions.

Usage

```
deviance_mixture(x, M, N, s, n, nbme1)
```

Arguments

x	value to which the deviance is to be evaluated
M	a vector of mixtures (see <code>coef_mixtures.R</code>)
N	a vector of bases (see <code>coef_mixtures.R</code>)
s	number of bases
n	number of cell probabilities
nbme1	number of mixtures

Value

This function returns the value of the deviance for mixture distributions.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

References

Yantis, S., Meyer, D. E., and Smith, J. E. K. (1991). Analyses of multinomial mixture distributions: New tests for stochastic models of cognition and action. *Psychological Bulletin* 110, 350–374.

expval_table	<i>Expected values in a contingency table</i>
--------------	---

Description

This function calculates expected values for a rxc contingency table.

Usage

```
expval_table(M)
```

Arguments

M a matrix of observed probabilities

Value

A matrix of expected values.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

gof_test	<i>Goodness-of-fit test for contingency tables</i>
----------	--

Description

This function carries out goodness-of-fit tests for contingency tables from the power-divergence family.

Usage

```
gof_test(lambda, observes, theoriques)
```

Arguments

lambda parameter defining the statistic to be used: lambda = -0.5 is for the Freeman-Tuckey statistic, lambda = 0 for the G2 statistic, lambda = 2/3 for the Cressie-Read statistic and lambda = 1 for the classical Chi-square statistic

observes vector of observed probabilities

theoriques vector of theoretical/expected probabilities

Value

This function returns the value of the goodness-of-fit statistic.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

group_data

Group individual capture-recapture data in encounter histories

Description

This function pools together individuals with the same encounter capture-recapture history.

Usage

```
group_data(X, effX)
```

Arguments

`X` matrix of capture-recapture histories
`effX` vector with numbers of individuals with that particular capture-recapture history

Value

matrix with grouped capture-recapture histories and counts in the last column

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

Examples

```
# Generate fake capture-recapture dataset  
X = matrix(round(runif(300)), nrow=100)  
freq=rep(1,100)  
cbind(X, freq)  
group_data(X, freq)
```

group_data_gen	<i>Group individual capture-recapture data in encounter histories along specific column(s)</i>
----------------	--

Description

This function pools together individuals with the same encounter capture-recapture history along specified directions given by columns.

Usage

```
group_data_gen(X, effX, s)
```

Arguments

X	matrix of capture-recapture histories
effX	vector with numbers of individuals with that particular capture-recapture history
s	scalar or vector of columns along which the grouping should be done

Value

matrix with grouped capture-recapture histories and counts in the last column

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

ind_test_22	<i>Test of independence for 2x2 contingency tables</i>
-------------	--

Description

This function tests independence in 2x2 contingency tables

Usage

```
ind_test_22(M, threshold = 2, rounding = 3)
```

Arguments

M	is a 2x2 contingency table
threshold	is a threshold for low expected numbers; default is 2
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a vector with statistic of quadratic chi2 or inv chi2 corresponding to pvalue of Fisher test, p-value of quadratic chi2 test or Fisher test for low numbers, signed test and test performed (Chi-square, Fisher or None).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

ind_test_rc *Test of independence for rxc contingency tables*

Description

This function tests independence in rxc contingency tables

Usage

```
ind_test_rc(M, threshold = 2, rounding = 3)
```

Arguments

M	is an r by c table of non-negative integers
threshold	is a threshold for low expected numbers; default is 2
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a vector with statistic of quadratic chi2 or inv chi2 corresponding to pvalue of Fisher test, p-value of quadratic chi2 test or Fisher test for low numbers, degree of freedom and test performed (Chi-square, Fisher or None).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

inv_logit_gen	<i>Inverse generalized logit link</i>
---------------	---------------------------------------

Description

This function computes the inverse (or reciprocal) of the generalized logit link function.

Usage

```
inv_logit_gen(petitv)
```

Arguments

petitv	vector of values to be transformed
--------	------------------------------------

Value

ev vector of transformed values

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

marray	<i>m-array: table of first recaptures</i>
--------	---

Description

This function calculates the m-array, the number of released and never seen again individuals; deals with more than 1 group

Usage

```
marray(X, freq)
```

Arguments

X	a matrix of encounter histories over K occasions
freq	is a vector with the number of individuals having the corresponding encounter history

Value

This function returns a list with R the number of released individuals (K-1 x g matrix), m the m-array (K-1 x K-1 x g array) with upper triangle filled only and never the number of individuals never recaptured (K-1 x g matrix).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

Examples

```
# read in the classical dipper dataset
dipper = system.file("extdata", "ed.inp", package = "R2ucare")
dipper = read_inp(dipper, group.df=data.frame(sex=c('Male', 'Female')))

# Get encounter histories, counts and groups:
dip.hist = dipper$encounter_histories
dip.freq = dipper$sample_size
dip.group = dipper$groups

# get female data
mask = (dip.group == 'Female')
dip.fem.hist = dip.hist[mask,]
dip.fem.freq = dip.freq[mask]

# get number of released individuals (R),
# the m-array (m) and
# the number of individuals never seen again (never)
marray(dip.fem.hist, dip.fem.freq)
```

multimarray

Multistate m-array

Description

This function calculates the m-array for multistate capture-recapture data, the number of released and never seen again individuals.

Usage

```
multimarray(X, freq)
```

Arguments

X	a matrix of encounter histories over K occasions
freq	is a vector with the number of individuals having the corresponding encounter history

Value

This function returns a matrix in which R the number of released individuals is in the first column, the number of individuals never recaptured (K-1) is in the last column and m the m-array (K-1 x K-1) with upper triangle filled only is in sandwich between these two vectors.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

Examples

```
# Read in Geese dataset:
geese = system.file("extdata", "geese.inp", package = "R2ucare")
geese = read_inp(geese)

# Get encounter histories and number of individuals with corresponding histories
geese.hist = geese$encounter_histories
geese.freq = geese$sample_size

# build m-array
multimarray(geese.hist, geese.freq)
```

overall_CJS

Overall goodness-of-fit test for the Cormack-Jolly-Seber model

Description

This function performs the overall goodness-of-fit test for the Cormack-Jolly-Seber model. It is obtained as the sum of the 4 components Test3.SR, Test3.SM, Test2.CT and Test2.CL.

Usage

```
overall_CJS(X, freq, rounding = 3)
```

Arguments

X	is a matrix of encounter histories
freq	is a vector of the number of individuals with the corresponding encounter history
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a data.frame with the value of the test statistic, the degrees of freedom and the p-value of the test.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

Examples

```

# read in the classical dipper dataset
dipper = system.file("extdata", "ed.inp", package = "R2ucare")
dipper = read_inp(dipper,group.df=data.frame(sex=c('Male','Female')))

# Get encounter histories, counts and groups:
dip.hist = dipper$encounter_histories
dip.freq = dipper$sample_size
dip.group = dipper$groups

# split the dataset in males/females
mask = (dip.group == 'Female')
dip.fem.hist = dip.hist[mask,]
dip.fem.freq = dip.freq[mask]
mask = (dip.group == 'Male')
dip.mal.hist = dip.hist[mask,]
dip.mal.freq = dip.freq[mask]

# for females
overall_CJS(dip.fem.hist, dip.fem.freq)

```

overall_JMV

Overall goodness-of-fit test for the Jolly-Move model

Description

This function performs the overall goodness-of-fit test for the Jolly-Move model. It is obtained as the sum of the 5 components Test3G.SR, Test3G.SM, Test3G.WBWA, TestM.ITEC, TestM.LTEC. To perform the goodness-of-fit test for the Arnason-Schwarz model, both the Arnason-Schwarz (AS) and the Jolly-Move models need to be fitted to the data (to our knowledge, only E-SURGE can fit the JMV model). Assuming the overall goodness-of-fit test for the JMV model has produced the value `stat_jmv` for the test statistic, get the deviance (say `dev_as` and `dev_jmv`) and number of estimated parameters (say `dof_as` and `dof_jmv`) for both the AS and JMV models. Then, finally, the p-value of the goodness-of-fit test for the AS model is obtained as $1 - \text{pchisq}(\text{stat_as}, \text{dof_as})$ where $\text{stat_as} = \text{stat_jmv} + (\text{dev_as} - \text{dev_jmv})$ and $\text{dof_as} = \text{dof_jmv} + (\text{dof_jmv} - \text{dof_as})$

Usage

```
overall_JMV(X, freq, rounding = 3)
```

Arguments

<code>X</code>	is a matrix of encounter histories
<code>freq</code>	is a vector of the number of individuals with the corresponding encounter history
<code>rounding</code>	is the level of rounding for outputs; default is 3

Value

This function returns a data.frame with the value of the test statistic, the degrees of freedom and the p-value of the test.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

Examples

```
# read in Geese dataset
library(RMark)
geese = system.file("extdata", "geese.inp", package = "R2ucare")
geese = convert.inp(geese)

geese.hist = matrix(as.numeric(unlist(strsplit(geese$ch, ' '))),nrow=nrow(geese),byrow=TRUE)
geese.freq = geese$freq

# encounter histories and number of individuals with corresponding histories
X = geese.hist
freq = geese.freq

# load R2ucare package
library(R2ucare)

# perform overall gof test
overall_JMV(X, freq)
```

pool2K

Pooling algorithm

Description

This function pools columns of a 2xK contingency table (if needed, ie if low numbers present)

Usage

```
pool2K(M, low = 2)
```

Arguments

M	is a 2 by K contingency table (or a K by 2 table)
low	is a threshold for low expected numbers; default is 2 (if this argument is big enough, the table is pooled down to 2 x 2; if this argument is 0, the table is not pooled)

Value

This function returns a matrix with the pooled contingency table.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

pooling_ct

Pooling algorithm (multisite goodness-of-fit tests)

Description

This function pools rows and columns of a rxc contingency table according to Pradel et al. (2003).

Usage

```
pooling_ct(table)
```

Arguments

table is a rxc contingency table

Value

This function returns a matrix with the pooled contingency table.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

References

Pradel R., Wintrebert C.M.A. and Gimenez O. (2003). A proposal for a goodness-of-fit test to the Arnason-Schwarz multisite capture-recapture model. *Biometrics* 59: 43-53.

pooling_mixtures	<i>Pooling algorithm (multisite goodness-of-fit tests)</i>
------------------	--

Description

This function pools rows and columns of a rxc bases and mixture table according to Pradel et al. (2003). It provides the components of TestM in the multisite goodness-of-fit tests.

Usage

```
pooling_mixtures(nk, nj, a, mixandbases)
```

Arguments

nk	number of mixtures
nj	number of bases
a	number of sites/states
mixandbases	matrix with mixtures and bases

Value

This function returns a matrix with the pooled table.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Rémi Choquet, Jean-Dominique Lebreton, Anne-Marie Reboulet, Roger Pradel

References

Pradel R., Wintrebert C.M.A. and Gimenez O. (2003). A proposal for a goodness-of-fit test to the Arnason-Schwarz multisite capture-recapture model. *Biometrics* 59: 43-53.

read_headed	<i>Read capture-recapture data with Headed format used by program E-SURGE</i>
-------------	---

Description

This function reads in capture-recapture dataset with the Headed format. It ignores all forms of censorship for now, and drops continuous covariates because no goodness-of-fit test exists for such models

Usage

```
read_headed(file)
```

Arguments

file text file with Headed format

Value

list with first component the matrix of encounter histories, second components the vector of number of individuals with corresponding histories and, if relevant, third component vector/matrix with group(s)

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>

Examples

```
# read in Dipper dataset
dipper = system.file("extdata", "ed.txt", package = "R2ucare")
read_headed(dipper)
# read in Geese dataset
geese = system.file("extdata", "geese.txt", package = "R2ucare")
read_headed(geese)
```

read_inp	<i>Read capture-recapture data with Input (.inp) format used by program MARK</i>
----------	--

Description

This function reads in capture-recapture dataset with the Input format. It is a wrapper for the function `convert.inp` from package `RMark`. It drops continuous covariates because no goodness-of-fit test exists for such models

Usage

```
read_inp(file, group.df = NULL)
```

Arguments

file text file with Input format (extension .inp)

group.df dataframe with grouping variables; contains a row for each group defined in the input file row1=group1, row2=group2 etc. Names and number of columns in the dataframe is set by user to define grouping variables in `RMark` dataframe

Value

list with first component the matrix of encounter histories, second components the vector of number of individuals with corresponding histories and, if relevant, third component vector/matrix with group(s)

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>

Examples

```
# read in Dipper dataset
dipper = system.file("extdata", "ed.inp", package = "R2ucare")
read_inp(dipper, group.df=data.frame(sex=c('Male', 'Female')))
# read in Geese dataset
geese = system.file("extdata", "geese.inp", package = "R2ucare")
read_inp(geese)
```

reconstitution

Reformat outputs of multinomial mixture distributions parameters

Description

This function reformat the outputs of multinomial mixture distributions parameters.

Usage

```
reconstitution(x, s, n, nbm1)
```

Arguments

x	vector with cell probabilities estimates for mixtures and bases, along with mixture probabilities
s	number of bases
n	number of cell probabilities
nbm1	number of mixtures

Value

This function returns a list of maximum likelihood estimates for the cells of a mixture distribution with:

P matrix of cell probabilities estimates for mixtures

PI matrix of mixture probabilities

GAM matrix of cell probabilities estimates for bases

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

repmat	<i>Replicate and tile a matrix</i>
--------	------------------------------------

Description

This function creates a large matrix consisting of an m-by-n tiling of copies of X. The dimensions of the returned matrix are $\text{nrow}(X)*m \times \text{ncol}(X)*n$. This is the equivalent of the repmat MATLAB function.

Usage

```
repmat(X, m, n)
```

Arguments

X	matrix to be replicated
m	row dimension of replication
n	column dimension of replication

Value

A replicated matrix of X with dimensions $\text{nrow}(X)*m \times \text{ncol}(X)*n$.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>

test2cl	<i>Test2.CL</i>
---------	-----------------

Description

This function performs Test2.CL

Usage

```
test2cl(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

X	is a matrix of encounter histories with K occasions
freq	is a vector of the number of individuals with the corresponding encounter history
verbose	controls the level of the details in the outputs; default is TRUE for all details
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with 5 columns for components i (2:K-3) (in rows) of test2.cli following Pradel 1993 (in Lebreton and North, Birkhauser Verlag): component, degree of freedom, statistic of the test, p-value, test performed.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

Examples

```
# read in the classical dipper dataset
dipper = system.file("extdata", "ed.inp", package = "R2ucare")
dipper = read_inp(dipper, group.df=data.frame(sex=c('Male', 'Female')))

# Get encounter histories, counts and groups:
dip.hist = dipper$encounter_histories
dip.freq = dipper$sample_size
dip.group = dipper$groups

# split the dataset in males/females
mask = (dip.group == 'Female')
dip.fem.hist = dip.hist[mask,]
dip.fem.freq = dip.freq[mask]
mask = (dip.group == 'Male')
dip.mal.hist = dip.hist[mask,]
dip.mal.freq = dip.freq[mask]

# for males
X = dip.mal.hist
freq = dip.mal.freq
res.males = test2cl(X, freq)
res.males
```

test2ct

Test2.CT

Description

This function performs Test2.CT

Usage

```
test2ct(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

X	is a matrix of encounter histories with K occasions
freq	is a vector of the number of individuals with the corresponding encounter history
verbose	controls the level of the details in the outputs; default is TRUE for all details
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with 5 columns for components i ($2:K-2$) (in rows) of test2.Cti: component, degree of freedom, statistic of the test, p-value, signed test, test performed.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

Examples

```
# read in the classical dipper dataset
dipper = system.file("extdata", "ed.inp", package = "R2ucare")
dipper = read_inp(dipper, group.df=data.frame(sex=c('Male', 'Female')))

# Get encounter histories, counts and groups:
dip.hist = dipper$encounter_histories
dip.freq = dipper$sample_size
dip.group = dipper$groups

# split the dataset in males/females
mask = (dip.group == 'Female')
dip.fem.hist = dip.hist[mask,]
dip.fem.freq = dip.freq[mask]
mask = (dip.group == 'Male')
dip.mal.hist = dip.hist[mask,]
dip.mal.freq = dip.freq[mask]

# for females
X = dip.fem.hist
freq = dip.fem.freq
res.females = test2ct(X, freq)
res.females
```

test3Gsm

Test3G.SM

Description

This function performs Test3G.SM

Usage

```
test3Gsm(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

X is a matrix of encounter histories with K occasions
freq is a vector of the number of individuals with the corresponding encounter history
verbose controls the level of the details in the outputs; default is TRUE for all details
rounding is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with occasion, site, the value of the test statistic, degree of freedom, p-value and test performed (chi-square, Fisher or none).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

Examples

```
# Read in Geese dataset:
geese = system.file("extdata", "geese.inp", package = "R2ucare")
geese = read_inp(geese)

# Get encounter histories and number of individuals with corresponding histories
geese.hist = geese$encounter_histories
geese.freq = geese$sample_size

# perform Test.3.GSm
test3Gsm(geese.hist, geese.freq)
```

test3Gsr

Test3G.SR

Description

This function performs Test3G.SR

Usage

```
test3Gsr(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

`X` is a matrix of encounter histories with `K` occasions
`freq` is a vector of the number of individuals with the corresponding encounter history
`verbose` controls the level of the details in the outputs; default is TRUE for all details
`rounding` is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with occasion, site, the value of the test statistic, degree of freedom, p-value and test performed (chi-square, Fisher or none).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Rémi Choquet, Roger Pradel

Examples

```
# Read in Geese dataset:
geese = system.file("extdata", "geese.inp", package = "R2ucare")
geese = read_inp(geese)

# Get encounter histories and number of individuals with corresponding histories
geese.hist = geese$encounter_histories
geese.freq = geese$sample_size

# perform Test3.GSR
test3Gsr(geese.hist, geese.freq)
```

test3Gwbwa

Test3G.WBWA

Description

This function performs Test3G.WBWA

Usage

```
test3Gwbwa(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

`X` is a matrix of encounter histories with `K` occasions
`freq` is a vector of the number of individuals with the corresponding encounter history
`verbose` controls the level of the details in the outputs; default is TRUE for all details
`rounding` is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with occasion, site, the value of the test statistic, degree of freedom, p-value and test performed (chi-square, Fisher or none).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

Examples

```
# Read in Geese dataset:
geese = system.file("extdata", "geese.inp", package = "R2ucare")
geese = read_inp(geese)

# Get encounter histories and number of individuals with corresponding histories
geese.hist = geese$encounter_histories
geese.freq = geese$sample_size

# perform Test.3GWBWA
test3Gwbwa(geese.hist,geese.freq)
```

test3sm

Test3.SM

Description

This function performs Test3.SM

Usage

```
test3sm(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

X	is a matrix of encounter histories with K occasions
freq	is a vector of the number of individuals with the corresponding encounter history
verbose	controls the level of the details in the outputs; default is TRUE for all details
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with 5 columns for components i (2:K-1) (in rows) of test3.smi: component, degree of freedom, statistic of the test, p-value, test performed.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

Examples

```
# read in the classical dipper dataset
dipper = system.file("extdata", "ed.inp", package = "R2ucare")
dipper = read_inp(dipper, group.df=data.frame(sex=c('Male', 'Female')))

# Get encounter histories, counts and groups:
dip.hist = dipper$encounter_histories
dip.freq = dipper$sample_size
dip.group = dipper$groups

# split the dataset in males/females
mask = (dip.group == 'Female')
dip.fem.hist = dip.hist[mask,]
dip.fem.freq = dip.freq[mask]
mask = (dip.group == 'Male')
dip.mal.hist = dip.hist[mask,]
dip.mal.freq = dip.freq[mask]

# for females
res.females = test3sm(dip.fem.hist, dip.fem.freq)
res.females
```

test3sr

Test3.SR

Description

This function performs Test3.SR

Usage

```
test3sr(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

X	is a matrix of encounter histories with K occasions
freq	is a vector of the number of individuals with the corresponding encounter history
verbose	controls the level of the details in the outputs; default is TRUE for all details
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with 4 columns for components i ($2:K-1$) (in rows) of test3.sri: component, statistic of the test, p-value, signed test, test performed.

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Jean-Dominique Lebreton, Rémi Choquet, Roger Pradel

Examples

```
# read in the classical dipper dataset
dipper = system.file("extdata", "ed.inp", package = "R2ucare")
dipper = read_inp(dipper, group.df=data.frame(sex=c('Male', 'Female')))

# Get encounter histories, counts and groups:
dip.hist = dipper$encounter_histories
dip.freq = dipper$sample_size
dip.group = dipper$groups

# split the dataset in males/females
mask = (dip.group == 'Female')
dip.fem.hist = dip.hist[mask,]
dip.fem.freq = dip.freq[mask]
mask = (dip.group == 'Male')
dip.mal.hist = dip.hist[mask,]
dip.mal.freq = dip.freq[mask]

# Test3SR for males
res.males = test3sr(dip.mal.hist, dip.mal.freq)
res.males
```

testMitec

TestM.ITEC

Description

This function performs TestM.ITEC

Usage

```
testMitec(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

X	is a matrix of encounter histories with K occasions
freq	is a vector of the number of individuals with the corresponding encounter history
verbose	controls the level of the details in the outputs; default is TRUE for all details
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with occasion, the value of the test statistic, degree of freedom, p-value and test performed (chi-square, Fisher or none).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Rémi Choquet, Roger Pradel

Examples

```
# Read in Geese dataset:
geese = system.file("extdata", "geese.inp", package = "R2ucare")
geese = read_inp(geese)

# Get encounter histories and number of individuals with corresponding histories
geese.hist = geese$encounter_histories
geese.freq = geese$sample_size

# perform TestM.ITEC
testMltec(geese.hist, geese.freq)
```

testMltec

TestM.LTEC

Description

This function performs TestM.LTEC

Usage

```
testMltec(X, freq, verbose = TRUE, rounding = 3)
```

Arguments

X	is a matrix of encounter histories with K occasions
freq	is a vector of the number of individuals with the corresponding encounter history
verbose	controls the level of the details in the outputs; default is TRUE for all details
rounding	is the level of rounding for outputs; default is 3

Value

This function returns a list with first component the overall test and second component a data.frame with occasion, the value of the test statistic, degree of freedom, p-value and test performed (chi-square, Fisher or none).

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

Examples

```
# Read in Geese dataset:
geese = system.file("extdata", "geese.inp", package = "R2ucare")
geese = read_inp(geese)

# Get encounter histories and number of individuals with corresponding histories
geese.hist = geese$encounter_histories
geese.freq = geese$sample_size

# perform TestM.LTEC
testMltec(geese.hist, geese.freq)
```

ungroup_data

Ungroup encounter capture-recapture data in individual histories

Description

This function splits encounter histories in as many individual histories as required.

Usage

```
ungroup_data(X, effX)
```

Arguments

X	matrix of encounter capture-recapture histories
effX	vector with numbers of individuals with that particular encounter history

Value

matrix with ungrouped capture-recapture histories and counts in the last column (should be 1s)

Author(s)

Olivier Gimenez <olivier.gimenez@cefe.cnrs.fr>, Roger Pradel, Rémi Choquet

Examples

```
# Generate fake capture-recapture dataset
X = matrix(round(runif(9)), nrow=3)
freq=c(4, 3, -8)
cbind(X, freq)
ungroup_data(X, freq)
```

Index

* package

- coef_mixture, 2
- deviance_mixture, 3
- expval_table, 4
- gof_test, 4
- group_data, 5
- group_data_gen, 6
- ind_test_22, 6
- ind_test_rc, 7
- inv_logit_gen, 8
- marray, 8
- multimarray, 9
- overall_CJS, 10
- overall_JMV, 11
- pool2K, 12
- pooling_ct, 13
- pooling_mixture, 14
- read_headed, 14
- read_inp, 15
- reconstitution, 16
- repmat, 17
- test2cl, 17
- test2ct, 18
- test3Gsm, 19
- test3Gsr, 20
- test3Gwbwa, 21
- test3sm, 22
- test3sr, 23
- testMitec, 24
- testMltec, 25
- ungroup_data, 26
- group_data_gen, 6
- ind_test_22, 6
- ind_test_rc, 7
- inv_logit_gen, 8
- marray, 8
- multimarray, 9
- overall_CJS, 10
- overall_JMV, 11
- pool2K, 12
- pooling_ct, 13
- pooling_mixture, 14
- read_headed, 14
- read_inp, 15
- reconstitution, 16
- repmat, 17
- test2cl, 17
- test2ct, 18
- test3Gsm, 19
- test3Gsr, 20
- test3Gwbwa, 21
- test3sm, 22
- test3sr, 23
- testMitec, 24
- testMltec, 25
- ungroup_data, 26

coef_mixture, 2

deviance_mixture, 3

expval_table, 4

gof_test, 4

group_data, 5