Package 'SparseMSE'

July 21, 2025

Title 'Multiple Systems Estimation for Sparse Capture Data'

Version 2.0.1

Author Lax Chan [aut, cre],
Bernard Silverman [aut],
Kyle Vincent [aut]

Maintainer Lax Chan <laxchan77@gmail.com>

Description Implements the routines and algorithms developed and analysed in ``Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists" Chan, L, Silverman, B. W., Vincent, K (2019) <doi:10.48550/arXiv.1902.05156>. This package explicitly handles situations where there are pairs of lists which have no observed individuals in common. It deals correctly with parameters whose estimated values can be considered as being negative infinity. It also addresses other possible issues of non-existence and non-identifiability of maximum likelihood estimates.

URL https://arxiv.org/abs/1902.05156

Depends R (>= 3.5.0)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

Imports lpSolve, Rcapture

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-15 12:30:02 UTC

Contents

`Artificial_3` *Artificial data set to demonstrate possible instabilities*

Description

This is a simple data set based on three lists, which gives examples of models that fail on one or the other of the criteria tested by `checkident`. This is Table 2 in Chan, Silverman and Vincent (2019).

Usage

```
Artificial_3
```

Format

An object of class `data.frame` with 4 rows and 4 columns.

Details

If all three two-list effects are included in the fitted model then the linear program in `checkident` yields a strictly positive value but the matrix A is not of full column rank, so the parameters are not identifiable. If the model contains AB either alone or in conjunction with one of AC and BC, then the linear program result is zero, so the MLE does not exist. If only main effects are considered, or if either or both of AC and BC, but not AB are included, then the model passes both tests.

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

`bcaconfvalues` *BCa confidence intervals*

Description

The BCa confidence intervals use percentiles of the bootstrap distribution of the population size , but adjust the percentile actually used. The adjusted percentiles depend on an estimated bias parameter, and the quantile function of the estimated bias parameter is the proportion of the bootstrap estimates that fall below the estimate from the original data, and an estimated acceleration factor, which derivation depends on a jackknife approach. This routine is called internally by `estimatepopulation`.

Usage

```
bcaconfvalues(bootreps, popest, ahat, alpha = c(0.025, 0.05, 0.1, 0.16,
  0.84, 0.9, 0.95, 0.975))
```

Arguments

`bootreps`	Point estimates of total population sizes from each bootstrap sample.
`popest`	A point estimate of the total population of the original data set.
`ahat`	the estimated acceleration factor
`alpha`	Bootstrap quantiles of interests

Value

BCa confidence intervals

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

DiCiccio, T. J. and Efron, B. (1996). Bootstrap Confidence Intervals. *Statistical Science*, **40(3)**, 189-228.

`BICandbootstrapsim` *Comparison of BIC approach and BCa approach*

Description

This routine carries out the simulation study as detailed in Section 3.4 of Chan, Silverman and Vincent (2019). If the original data set has low counts, so that there is a possibility of a simulated data set containing empty lists, then it may be advisable to use the option `noninformativelist=TRUE`.

Usage

```
BICandbootstrapsim(zdat, nsims = 100, nboot = 100, pthresh = 0.02,
  iseed = 1234, alpha = c(0.025, 0.05, 0.1, 0.16, 0.5, 0.84, 0.9, 0.95,
  0.975), noninformativelist = F, verbose = F, ...)
```

Arguments

`zdat`	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
`nsims`	Number of simulations to be carried out.
`nboot`	Number of bootstrap replications for each simulation
`pthresh`	p-value threshold used in `estimatepopulation.0`.
`iseed`	seed for initialization.
`alpha`	bootstrap quantiles of interests.
`noninformativelist`	if `noninformativelist=TRUE` then each generated data set in the simulation study (including all bootstrap replications) will be passed to `removenoninformativelists`.
`verbose`	If `verbose=FALSE`, then the progress of the simulation will not show. If `verbose=TRUE`, then the progress of the simulation will be shown.
`...`	other arguments.

Value

A list with components as below

`popest` Total population point estimate from the original data using `estimatepopulation.0` with default threshold.

`BICmodels` The best model chosen by the BIC at each simulation.

`BICvals` Point estimates of the total population and standard error of the best model chosen by the BIC at each simulation.

`simreps` Counts associated to each capture history at each simulation.

`modelmat` A full capture history matrix excluding the row corresponding to the dark figure.

`popestsim` Total population estimate given by the BCa method in each simulation.

`BCaquantiles` bootstrap confidence intervals given by the BCa method.

`BICconf` confidence interval given by the BIC method.

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

DiCiccio, T. J. and Efron, B. (1996). Bootstrap Confidence Intervals. *Statistical Science*, **40(3)**, 189-228.

Rivest, L-P. and Baillargeon, S. (2014) Rcapture. CRAN package. Available from Available from https://CRAN.R-project.org/package=Rcapture.

`buildmodel` *Build model for multiple systems estimation*

Description

For multiple systems estimation model corresponding to a specified set of two-list effects, set up the GLM model formula and data matrix.

Usage

```
buildmodel(zdat, mX)
```

Arguments

`zdat`	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero counts.
`mX`	A $2 \times k$ matrix giving the k two-list effects to be included in the model. Each column of `mX` contains the numbers of the corresponding pair of lists. If `mX = 0`, then all two-list effects are included. If `mX = NULL`, no such effects are included and the main effects model is fitted.

Value

A list with components as below.

`datamatrix` A matrix with all possible capture histories, other than those equal to or containing non-overlapping pairs indexed by parameters that are within the model specified by `mX`. A non-overlapping pair is a pair of lists (i, j) such that no case is observed in both lists, regardless of

whether it is present on any other lists. If (i, j) is within the model specified by `mX`, all capture histories containing both i and j are then excluded.

`modelform` The model formula suitable to be called by the Generalized Linear Model function `glm`. Model terms corresponding to non-overlapping pairs are not included, because they are handled by removing appropriate rows from the data matrix supplied to glm. The list of non-overlapping pairs are provided in `emptyoverlaps`. See Chan, Silverman and Vincent (2019) for details.

`emptyoverlaps` A matrix with two rows, whose columns give the indices of non-overlapping pairs of lists where the parameter indexed by the pair is within the specified model. The column names give the names of the lists corresponding to each pair.

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

Examples

```
data(NewOrl)
buildmodel(NewOrl, mX=NULL)
#Build a matrix that contains all two-list effects
m=dim(Artificial_3)[2]-1
mX = t(expand.grid(1:m, 1:m)); mX = mX[ , mX[1,]<mX[2,]]
# With one two-list effect
buildmodel(NewOrl, mX=mX[,1])
#With three two-list effects
buildmodel(NewOrl, mX=mX[,1:3])
```

`buildmodelmatrix` *Build the model matrix based on particular data, as required to check for identifiability and existence of the maximum likelihood estimate*

Description

This routine builds a model matrix as required by the linear program check `checkident` and checks if the matrix is of full rank. In addition, for each individual list, and for each pair of lists included in the model, it returns the total count of individuals appearing on the specific list or lists whether or not in combination with other lists.

Usage

```
buildmodelmatrix(zdat, mX = NULL)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
mX	A $2 \times k$ matrix giving the k two-list parameters to be included in the model. Each column of mX contains the numbers of the corresponding pair of lists. If mX = 0, then all two-list parameters are included. If mX = NULL, no such parameters are included and the main effects model is fitted.

Value

A list with components as below

modmat The matrix that maps the parameters in the model (excluding any corresponding to non-overlapping lists) to the log expected value of the counts of capture histories that do not contain non-overlapping pairs in the data.

tvec A vector indexed by the parameters in the model, excluding those corresponding to non-overlapping pairs of lists. For each parameter the vector contains the total count of individuals in all the capture histories that include both the lists indexed by that parameter.

rankdef The column rank deficiency of the matrix modmat. If rankdef = 0, the matrix has full column rank.

Examples

```
data(NewOrl)
buildmodelmatrix(NewOrl, mX=NULL)
```

---

checkallmodels *Check all possible models for existence and identifiability*

Description

This routine efficiently checks whether every possible model satisfies the conditions for the existence and identifiability of an extended maximum likelihood estimate. For identifiability it is only necessary to check the model containing all two-list effects. For existence, the approach set out in Chan, Silverman and Vincent (2019) is used. This uses the linear programming approach described in a more general and abstract context by Fienberg and Rinaldo (2012).

Usage

```
checkallmodels(zdat, nreport = 1024)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
nreport	A message is printed for every nreport models checked. It gives the number of top level models to be checked altogether, and also the number of models found so far for which the estimate does not exist.

Details

If the extended maximum likelihood estimator exists for a particular model, then it will still exist if one or more overlapping pairs are removed from the model. This allows the search to be carried out efficiently, as follows:

1. Search over 'top-level' models which contain all overlapping pairs. The number of such models is 2^k, where k is the number of non-overlapping pairs, because there are 2^k possible choices of the set of non-overlapping pairs to include in the model.

2. For each top-level model, if the estimate exists there is no need to consider that model further. If the estimate does not exist, then a branch search is carried out over the tree structure obtained by successively leaving out overlapping pairs.

Value

The routine prints a message if the model with all two-list parameters is not identifiable. As set out above, it gives regular progress reports in the case where there are a large number of models to be checked.

If all models give estimates which exist, then a message is printed to that effect.

If there are models for which the estimate does not exist, the routine reports the number of such models and returns a matrix whose rows give the parameters included in them.

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

Fienberg, S. E. and Rinaldo, A. (2012). Maximum likelihood estimation in log-linear models. *Ann. Statist.* **40**, 996-1023. Supplementary material: Technical report, Carnegie Mellon University. Available from http://www.stat.cmu.edu/~arinaldo/Fienberg_Rinaldo_Supplementary_Material.pdf.

Examples

```
data(Artificial_3)
data(Western)
checkallmodels(Artificial_3)
checkallmodels(Western)
```

checkident *Check a model for the existence and identifiability of the maximum likelihood estimate*

Description

Apply the linear programming test as derived by Fienberg and Rinaldo (2012), and a calculation of the rank of the design matrix, to check whether a particular model yields an identifiable maximum likelihood estimate based on the given data. The linear programming problem is as described on page 3 of Fienberg and Rinaldo (2012), with a typographical error corrected. Further details are given by Chan, Silverman and Vincent (2019).

Usage

```
checkident(zdat, mX = 0, verbose = FALSE)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has observed count zero.
mX	A $2 \times k$ matrix giving the k two-list parameters to be included in the model. Each column of mX contains the numbers of the corresponding pair of lists. If mX = 0, then all two-list interactions are included. If mX = NULL, no two-list parameters are included and the main effects model is fitted.
verbose	Specifies the output. If FALSE then the error code is returned. If TRUE then in addition the routine prints an error message if the model/data fail either of the two tests, and also returns both the error code and the lp object.

Value

If verbose=FALSE, then return the error code ierr which is 0 if there are no errors, 1 if the linear program test shows that the maximum likelihood estimate does not exist, 2 if it is not identifiable, and 3 if both tests are failed.

If verbose=TRUE, then return a list with components as below

ierr As described above.

zlp Linear programming object, in particular giving the value of the objective function at optimum.

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

Fienberg, S. E. and Rinaldo, A. (2012). Maximum likelihood estimation in log-linear models. *Ann. Statist.* **40**, 996-1023. Supplementary material: Technical report, Carnegie Mellon University. Available from http://www.stat.cmu.edu/~arinaldo/Fienberg_Rinaldo_Supplementary_Material.pdf.

Examples

```
data(Artificial_3)
#Build a matrix that contains all two-list effects
m=dim(Artificial_3)[2]-1
mX = t(expand.grid(1:m, 1:m)); mX = mX[ , mX[1,]<mX[2,]]
# When the model is not identifiable
checkident(Artificial_3,mX=mX, verbose=TRUE)
# When the maximum likelihood estimate does not exist
checkident(Artificial_3, mX=mX[,1],verbose=TRUE)
#Model passes both tests
checkident(Artificial_3, mX=mX[,2:3],verbose=TRUE)
```

---

checkthetasubset *Check a subset of the parameter set theta*

Description

This routine leaves out a particular set of parameters corresponding to the two-list effects from the parameter set theta. For the resulting model, it constructs the linear programming problem to check whether the extended maximum likelihood estimates of the parameters exists. It is called internally by `checkallmodels`.

Usage

```
checkthetasubset(zset, amat, tvec, nlists)
```

Arguments

`zset`	set of indices that is not included, numbered among the two-list effects only
`amat`	a design matrix
`tvec`	vector of sufficient statistics
`nlists`	number of lists in the original capture-recapture matrix

Value

If the return result is `TRUE`, the linear program shows that the extended maximum likelihood estimate does not exist. If the return result is `FALSE`, the estimate exists.

count_triples *Count number of triples of overlapping lists*

Description

The routine counts the number of subsets of size three of lists such that every pair of lists in the triple overlaps. If the number is zero, then the model with all two-list effects is unidentifiable.

Usage

```
count_triples(zdat)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.

Value

a count of subsets of size three of lists such that every pair of lists in the triple overlaps.

Examples

```
data(Western)
data(Artificial_3)
count_triples(Western)
count_triples(Artificial_3)
```

estimatepopulation *Bootstrapping to evaluate confidence intervals using BCa methods*

Description

This routine implements the bootstrapping and jacknife approach as detailed in Section 3.3 of Chan, Silverman and Vincent (2019). It calls the routine `estimatepopulation.0` and so is the preferred routine to be called if a user wishes to estimate the population and obtain BCa confidence intervals.

Usage

```
estimatepopulation(zdat, nboot = 1000, pthresh = 0.02, iseed = 1234,
  alpha = c(0.025, 0.05, 0.1, 0.16, 0.84, 0.9, 0.95, 0.975), ...)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
nboot	Number of bootstrap replications.
pthresh	p-value threshold used in estimatepopulation.0.
iseed	seed for initialisation.
alpha	Bootstrap quantiles of interests.
...	other arguments which will be passed to estimatepopulation.0

Value

A list with components as below:

popest point estimate of the total population of the original data set

MSEfit model fitted to the data, in the format described in modelfit

bootreps point estimates of total population sizes from each bootstrap sample

ahat the estimated acceleration factor

BCaquantiles BCa confidence intervals

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

DiCiccio, T. J. and Efron, B. (1996). Bootstrap Confidence Intervals. *Statistical Science*, **40(3)**, 189-228.

---

estimatepopulation.0 *Estimate the total population including the dark figure. If the user wishes to find bootstrap confidence intervals then the routine estimatepopulation should be used instead.*

Description

This routine estimates the total population size, which includes the dark figure, together with confidence intervals as specified. It also returns the details of the fitted model. The user can choose whether to fit main effects only, to fit a particular model containing specified two-list parameters, or to choose the model using the stepwise approach described by Chan, Silverman and Vincent (2019).

Usage

```
estimatepopulation.0(zdat, method = "stepwise", quantiles = c(0.025,
  0.975), mX = NULL, pthresh = 0.02)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
method	If method = "stepwise" the stepwise method implemented in stepwisefit is used. If method = "fixed" then a specified fixed model is used; the model is then given by mX. If method = "main" then main effects only are fitted.
quantiles	Quantiles of interest for confidence intervals.
mX	A $2 \times k$ matrix giving the k two-list parameters to be included in the model if method = "fixed". Each column of mX contains the numbers of the corresponding pair of lists. If mX = 0, then all two-list parameters are included. If mX = NULL, no two-list parameters are included and the main effects model is fitted. If only one two-list parameter is to be fitted, it is sufficient to specify it as a vector of length 2, e.g mX=c(1,3) for the parameter indexed by lists 1 and 3. If method is equal to "stepwise" or "main", then mX is ignored.
pthresh	Threshold p-value used if method = "stepwise".

Value

A list with components as below

estimate Point estimate and confidence interval estimates corresponding to specified quantiles.

MSEfit The model fitted to the data in the format described in modelfit.

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

Examples

```
data(NewOrl)
data(NewOrl_5)
estimatepopulation.0(NewOrl, method="stepwise", quantiles=c(0.025,0.975))
estimatepopulation.0(NewOrl_5, method="main", quantiles=c(0.01, 0.05,0.95, 0.99))
```

---

investigateAIC *Plot of simulation study*

Description

This routine reproduces Figure 1 of Chan, Silverman and Vincent (2019).

Usage

```
investigateAIC(nsim = 10000, Nsamp = 1000, seed = 1001)
```

Arguments

nsim	The number of simulation replications
Nsamp	The expected value of the total population size within each simulation
seed	The random number seed

Details

Simulations are carried out for two different three-list models. In one model, the probabilities of capture are 0.01, 0.04 and 0.2 for the three lists respectively, while in the other the probability is 0.3 on all three lists. In both cases, captures on the lists occur independently of each other, so there are no two-list effects. The first model is chosen to be somewhat more typical of the sparse capture case, of the kind which often occurs in the human trafficking context, while the second, more reminiscent of a classical mark-recapture study.

The probability of an individual having each possible capture history is first evaluated. Then these probabilities are multiplied by `Nsamp = 1000` and, for each simulation replicate, Poisson random values with expectations equal to these values are generated to give a full set of observed capture histories; together with the null capture history the expected number of counts (population size) is equal to `Nsamp`. Inference was carried out both for the model with main effects only, and for the model with the addition of a correlation effect between the first two lists. The reduction in deviance between the two models was determined. Ten thousand simulations were carried out.

Checking for compliance with the conditions for existence and identifiability of the estimates shows that a very small number of the simulations for the sparse model (two out of ten thousand) fail the checks for existence even within the extended maximum likelihood context. Detailed investigation shows that in neither of these cases is the dark figure itself not estimable; although the parameters themselves cannot all be estimated, there is a maximum likelihood estimate of the expected capture frequencies, and hence the deviance can still be calculated.

The routine produces QQ-plots of the resulting deviance reductions against quantiles of the χ^2_1 distribution, for `nsim` simulation replications.

Value

An `nsim` $\times$ 2 matrix giving the changes in deviance for each replication for each of the two models.

References

Chan, L., Silverman, B. W., and Vincent, K. (2019). Multiple Systems Estimation for Sparse Capture Data: Inferential Challenges when there are Non-Overlapping Lists. Available from https://arxiv.org/abs/1902.05156.

modelfit *Fit a specified model to multiple systems estimation data*

Description

This routine fits a specified model to multiple systems estimation data, taking account of the possibility of empty overlaps between pairs of observed lists.

Usage

```
modelfit(zdat, mX = NULL, check = TRUE)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
mX	A $2 \times k$ matrix giving the k two-list parameters to be included in the model. Each column of mX contains the numbers of the corresponding pair of lists. If mX = 0, then all two-list parameters are included. If mX = NULL, no two-list parameters are included and the main effects model is fitted. If only one two-list parameter is to be fitted, it may be specified as a vector of length 2, e.g mX=c(1,3) for the parameter corresponding to lists 1 and 3.
check	If check = TRUE check first of all if the maximum likelihood estimate exists and is identifiable, using the routine checkident. If either condition fails, print an appropriate error message and return the error code.

Value

A list with components as below

fit Details of the fit of the specified model as output by glm. The Akaike information criterion is adjusted to take account of the number of parameters corresponding to non-overlapping pairs.

emptyoverlaps Matrix with two rows, giving the list pairs within the model for which no cases are observed in common. Each column gives the indices of a pair of lists, with the names of the lists in the column name.

poisspempty the Poisson p-values of the parameters corresponding to non-overlapping pairs.

Examples

```
data(NewOrl)
modelfit(NewOrl,mX= c(1,3), check=TRUE)
```

Ned *The Netherlands data*

Description

Victims related to human trafficking in the Netherlands

Usage

```
Ned
```

Format

An object of class `data.frame` with 24 rows and 7 columns.

Details

These data are collected into six lists. Full details are given in Table 2. of Silverman (2019).

References

Silverman, B. W. (2019). Model fitting in Multiple Systems Analysis for the quantification of Modern Slavery: Classical and Bayesian approaches *Journal of Royal Statistical Society: Series A to appear*.

Ned_5 *Netherlands data five list version*

Description

Netherlands data consolidated into five lists

Usage

```
Ned_5
```

Format

An object of class `data.frame` with 17 rows and 6 columns.

Details

This reduces the Netherlands data `Ned` into five lists, constructed by combining the two smallest lists I and O into a single list.

NewOrl *New Orleans data*

Description

Victims related to human trafficking in Greater New Orleans

Usage

```
NewOrl
```

Format

An object of class `data.frame` with 19 rows and 9 columns.

Details

These data are collected into 8 lists. For reasons of confidentiality the lists are only labelled as A, B, ..., H. Full details are given in Bales, Murphy and Silverman (2019).

References

K. Bales, L. Murphy and B. W. Silverman (2019). How many trafficked people are there in Greater New Orleans? *Journal of Human Trafficking, to appear*.

NewOrl_5 *New Orleans data five list version*

Description

New Orleans data consolidated into five lists

Usage

```
NewOrl_5
```

Format

An object of class `data.frame` with 14 rows and 6 columns.

Details

This reduces the New Orleans data `NewOrl` into five lists, constructed by combining the four smallest lists B, E, F and G into a single list.

`ordercaptures` *Order capture histories*

Description

Given a matrix with capture histories only, the routine orders the capture histories first by the number of 1s in the capture history and then lexicographically by columns.

Usage

```
ordercaptures(zmat)
```

Arguments

`zmat`	Data matrix with t columns. The t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. Where a capture history is not explicitly listed, it is assumed that it has observed count zero.

Value

A data matrix that is ordered first by the number of 1s in the capture history and then lexicographically by columns.

`removenoninformativelists` *Remove non-informative list*

Description

The routine cleans up the data set by removing any lists that contain no data, any lists which contain all the observed data, and any list whose results duplicate those of another list. If as a result the original data set has no list left, it returns a matrix with the value of the total count.

Usage

```
removenoninformativelists(zdat)
```

Arguments

`zdat`	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history.

Value

data matrix that contains no duplicate lists, no lists with no data, and no lists that contain all the observed data. If all lists are removed, the total count is returned.

`stepwisefit` *Stepwise fit using Poisson p-values.*

Description

Starting with a model with main effects only, two-list parameters are added one by one. At each stage the parameter with the lowest p-value is added, provided that p-value is lower than `pthresh`, and provided that the resulting model does not fail either of the tests in `checkident`.

Usage

```
stepwisefit(zdat, pthresh = 0.02)
```

Arguments

`zdat`	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
`pthresh`	this is the threshold below which the p-value of the newly added parameter needs to be in order to be included in the model. If `pthresh = 0` then the model with main effects only is returned.

Details

For each candidate two-list parameter for possible addition to the model, the p-value is calculated as follows. The total of cases occurring on both lists indexed by the parameter (regardless of whether or not they are on any other lists) is calculated. On the null hypothesis that the effect is not included in the model, this statistic has a Poisson distribution whose mean depends on the parameters within the model. The one-sided Poisson p-value of the observed statistic is calculated.

Value

A list with components as below

`fit` Details of the fit of the specified model as output by `glm`. The Akaike information criterion is adjusted to take account of the number of parameters corresponding to non-overlapping pairs.

`emptyoverlaps` Matrix with two rows, giving the list pairs within the model for which no cases are observed in common. Each column gives the indices of a pair of lists, with the names of the lists in the column name.

`poisspempty` the Poisson p-values of the parameters corresponding to non-overlapping pairs.

Examples

```
data(NewOrl)
stepwisefit(NewOrl, pthresh=0.02)
```

subsetsearch *Search subsets for a property which is inherited in a particular way*

Description

The following routine returns a list of all subsets of a given set for which a specified property is TRUE. It is assumed that if the property is FALSE for any particular subset, it is also FALSE for all supersets of that subset. This enables a branch search strategy to be used to obviate the need to search over supersets of subsets already eliminated from consideration. It is used within the hierarchical search step of checkallmodels.

Usage

```
subsetsearch(n, checkfun, testnull = TRUE, ...)
```

Arguments

n	an integer such that the search is over all subsets of $\{1, ..., n\}$
checkfun	a function which takes arguments zset, a subset of $\{1, ..., n\}$ and ... The function returns the value TRUE or FALSE. It needs to have the property that if it is FALSE for any particular zset, it is also FALSE for all supersets of zset.
testnull	If TRUE, then include the null subset in the search. If FALSE, do not test the null subset.
...	other arguments

Value

A list of all subsets for which the value is TRUE

tidylists *Produce a data matrix with a unique row for each capture history*

Description

This routine finds rows with the same capture history and consolidates them into a single row whose count is the sum of counts of the relevant rows. If includezerocounts = TRUE then it also includes rows for all the capture histories with zero count; otherwise these are all removed.

Usage

```
tidylists(zdat, includezerocounts = FALSE)
```

Arguments

zdat	Data matrix with $t + 1$ columns. The first t columns, each corresponding to a particular list, are 0s and 1s defining the capture histories observed. The last column is the count of cases with that particular capture history. List names A, B, ... are constructed if not supplied. Where a capture history is not explicitly listed, it is assumed that it has zero count.
includezerocounts	If FALSE then remove rows corresponding to capture histories with zero count. If TRUE then include all possible capture histories including those with zero count, excluding the all-zero row corresponding to the dark figure.

Value

A data matrix in the form specified above, including all capture histories with zero counts if includezerocounts=TRUE.

Examples

```
data(NewOrl)
zdat<-tidylists(NewOrl,includezerocounts=TRUE)
```

UKdat *UK data*

Description

Data from the UK 2013 strategic assessment

Usage

```
UKdat
```

Format

An object of class data.frame with 25 rows and 7 columns.

Details

This is a table of six lists used in the research published by the Home Office as part of the strategy leading to the Modern Slavery Act 2015. The data are considered in six lists, labelled as follows: LA–Local authorities; NG–Non-government organisations; PF–Police forces; GO–Government organisations; GP–General public; NCA–National Crime Agency. Each of the first six columns in the data frame corresponds to one of these lists. Each of the rows of the data frame corresponds to a possible combination of lists, with value 1 in the relevant column if the list is in that particular combination. The last column of the data frame states the number of cases observed in that particular combination of lists. Combinations of lists for which zero cases are observed are omitted.

References

https://www.gov.uk/government/publications/modern-slavery-an-application-of-multiple-systems-estima

---

`UKdat_5` *UK data five list version*

Description

UK data consolidated into five lists

Usage

```
UKdat_5
```

Format

An object of class `data.frame` with 18 rows and 6 columns.

Details

This reduces the UK data `UKdat` into five lists, constructed by combining the PF and NCA lists into a single PFNCA list

---

`Western` *Victims related to sex trafficking in a U.S. Western site*

Description

These data are collected into 5 lists. For reasons of confidentiality the lists are only labelled as A, B, C, D and E. Full details are given in Farrell, Dank, Kfafian, Lockwood, Pfeffer, Hughes and Vincent (2019).

Usage

```
Western
```

Format

An object of class `data.frame` with 13 rows and 6 columns.

References

Farrell, A., Dank, M., Kfafian, M., Lockwood, S., Pfeffer, R., Hughes, A., and Vincent, K. (2019). Capturing human trafficking victimization through crime reporting. Technical Report 2015-VF-GX-0105, National Institute of Justice. Available from https://www.ncjrs.gov/pdffiles1/nij/grants/252520.pdf.

Index