

# Package ‘MCARtest’

July 17, 2023

**Title** Optimal Nonparametric Testing of Missing Completely at Random

**Version** 1.1

**Author** Thomas B. Berrett <tom.berrett@warwick.ac.uk> [aut,cre] (0000-0002-2005-110X), Danat Duisen-bekov <dd583@cam.ac.uk> [aut], Sean Jaffe <scj47@cam.ac.uk> [aut], Richard J. Samworth <r.samworth@statslab.cam.ac.uk> [aut] (0000-0003-2426-4679)

**Maintainer** Thomas B. Berrett <tom.berrett@warwick.ac.uk>

**Description** Provides functions for carrying out nonparametric hypothesis tests of the MCAR hypothesis based on the theory of Frechet classes and compatibility. Also gives functions for computing halfspace representations of the marginal polytope and related geometric objects.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Imports** IpSolve, rcd, gtools, Epi, Rdpack, Rcpp, Matrix, highs

**LinkingTo** Rcpp

**RdMacros** Rdpack

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2023-07-17 08:10:03 UTC

## R topics documented:

Amatrix . . . . .	2
AmatrixSparse . . . . .	3
Cimproved . . . . .	3
col_index . . . . .	4
ConsMinkSumHrep . . . . .	5
Csimple . . . . .	6
EMiteration . . . . .	7
EquivalenceClass . . . . .	8
FuchsTest . . . . .	8

InconsMinkSumHrep . . . . .	10
loglik0 . . . . .	11
loglik1 . . . . .	12
MargPolyHrep . . . . .	13
MLE . . . . .	14
ProjectionTest . . . . .	15
Rindex . . . . .	16
RindexDual . . . . .	17
RoundErrors . . . . .	18
row_index . . . . .	19
<b>Index</b>	<b>20</b>

---

Amatrix	<i>Generate the matrix A, whose columns are the vertices of the marginal polytope.</i>
---------	--

---

### Description

Generate the matrix A, whose columns are the vertices of the marginal polytope.

### Usage

```
Amatrix(bS, M)
```

### Arguments

bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.

### Value

The matrix A.

### Examples

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
M=c(2,2,2)
Amatrix(bS,M)
```

---

AmatrixSparse	<i>Generate the matrix A, whose columns are the vertices of the marginal polytope, as a sparse matrix.</i>
---------------	--

---

**Description**

Generate the matrix A, whose columns are the vertices of the marginal polytope, as a sparse matrix.

**Usage**

```
AmatrixSparse(bS, M)
```

**Arguments**

bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.

**Value**

The matrix A.

**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
M=c(2,2,2)
AmatrixSparse(bS,M)
```

---

Cimproved	<i>Calculate the critical value for our improved test</i>
-----------	---

---

**Description**

Calculate a critical value for an MCAR test based on knowledge of the facet structure of the Minkowski sum calculated by ConsMinkSumHrep.

**Usage**

```
Cimproved(nS, bS, M, DR, Fp, alpha)
```

**Arguments**

nS	A vector of sample sizes, with each entry corresponding to an observation pattern.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
DR	The quantity $D_R$ appearing in Berrett and Samworth (2022).
Fp	The quantity $F'$ appearing in Berrett and Samworth (2022).
alpha	The desired significance level $\alpha$ of the test.

**Value**

The critical value  $C'_\alpha$  defined in Berrett and Samworth (2022).

**References**

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.

**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
r=4; s=3
M=c(r,s,2)
Cimproved(rep(1000,3),bS,M,1,(2^r-2)*(2^s-2),0.05)
```

---

col\_index

*A function indexing the columns of A*


---

**Description**

A map from the joint space to an index set.

**Usage**

```
col_index(M, x)
```

**Arguments**

M	A vector of positive integers giving the alphabet sizes of the discrete variables.
x	An element of the joint space.

**Value**

A positive integer no greater than the cardinality of the joint space uniquely identifying x.

**Examples**

```
M=c(2,2,2)
col_index(M,c(1,1,1))
col_index(M,c(1,1,2))
```

```
M=c(4,3,2)
col_index(M,c(1,1,1))
col_index(M,c(2,1,1))
col_index(M,c(1,2,1))
col_index(M,c(1,1,2))
```

---

ConsMinkSumHrep

*Calculate the H-representation of the consistent Minkowski sum*


---

**Description**

Computes the minimal halfspace representation of the Minkowski sum of the marginal polytope and the consistent ball defined in Berrett and Samworth (2022).

**Usage**

```
ConsMinkSumHrep(bS, M, round = FALSE)
```

**Arguments**

bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
round	A logical value indicating whether or not to round coefficients to 15 significant figures. The function RoundErrors can be used separately to substitute other values for 15. Defaults to FALSE.

**Value**

A halfspace representation object as used by the rcdd package. See Geyer and Meeden (2021) for more detail.

**References**

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.

Geyer CJ, Meeden GD (2021). *rcdd: Computational Geometry*. <https://CRAN.R-project.org/package=rcdd>.

**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
ConsMinkSumHrep(bS,c(2,2,2))
```

---

Csimple

*Calculate the critical value for our simple test*


---

### Description

Calculate a simple critical value for an MCAR test using only knowledge of the set of observation patterns and the joint observation space.

### Usage

```
Csimple(nS, bS, M, alpha)
```

### Arguments

nS	A vector of sample sizes, with each entry corresponding to an observation pattern.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
alpha	The desired significance level $\alpha$ of the test.

### Value

The universal critical value defined in Berrett and Samworth (2022).

### References

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.

### Examples

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
r=4; s=3
M=c(r,s,2)
Csimple(rep(1000,3),bS,M,0.05)
```

---

EMiteration	<i>Perform one step of the EM algorithm for finding the MLE under MCAR in a contingency table.</i>
-------------	--

---

**Description**

Perform one step of the EM algorithm for finding the MLE under MCAR in a contingency table.

**Usage**

```
EMiteration(pt, p0h, n0, pSh, nS, bS, M)
```

**Arguments**

pt	An input probability mass function on the joint space, to be updated.
p0h	An empirical mass function calculated using complete observations.
n0	An integer giving the number of complete observations used to calculate p0h.
pSh	A sequence of empirical mass functions calculated using incomplete observations.
nS	A sequence of integers giving the numbers of incomplete observations used to calculate pSh.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.

**Value**

The updated probability mass function on the joint space.

**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3) # Our canonical 3d example
M=c(2,2,2)
n0=200
nS=c(200,200,200)

pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
P12=pS[1:4]; P13=pS[5:8]; P23=pS[9:12]
X12=t(rmultinom(1,size=nS[1],prob=P12)/nS[1])
X13=t(rmultinom(1,size=nS[2],prob=P13)/nS[2])
X23=t(rmultinom(1,size=nS[3],prob=P23)/nS[3])
pSh=cbind(X12,X13,X23)

p0=array(0.125,dim=c(2,2,2))
p0h=array(rmultinom(1,n0,p0),dim=M)/n0

EMiteration(p0,p0h,n0,pSh,nS,bS,M)
```

---

EquivalenceClass	<i>Simplifies H-representation by exploiting symmetry</i>
------------------	---

---

### Description

The marginal polytope and related objects have many symmetries. By relabelling the levels of discrete variables we transform facets into other facets. This function reduces a list of halfspace normals to its equivalence classes.

### Usage

```
EquivalenceClass(bS, M, Hrep)
```

### Arguments

bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
Hrep	An H-representation generated by MargPolyHrep, ConsMinkSumHrep or InconsMinkSumHrep.

### Value

A list of representative halfspace normals.

### Examples

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3) # Our canonical 3d example
Hrep=MargPolyHrep(bS,c(2,2,2))
EquivalenceClass(bS,c(2,2,2),Hrep)
```

---

FuchsTest	<i>Carry out Fuchs's test of MCAR in a contingency table, given complete and incomplete observations.</i>
-----------	---

---

### Description

Carry out Fuchs's test of MCAR in a contingency table, given complete and incomplete observations.

### Usage

```
FuchsTest(p0h, n0, pSh, nS, bS, M, Niter)
```



**Arguments**

<code>p0h</code>	An empirical mass function calculated using complete observations.
<code>n0</code>	An integer giving the number of complete observations used to calculate <code>p0h</code> .
<code>pSh</code>	A sequence of empirical mass functions calculated using incomplete observations.
<code>nS</code>	A sequence of integers giving the numbers of incomplete observations used to calculate <code>pSh</code> .
<code>bS</code>	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
<code>M</code>	A vector of positive integers giving the alphabet sizes of the discrete variables.
<code>Niter</code>	An integer giving the number of iterations to be used in the EM algorithm for calculating the null MLE.

**Value**

The p-value of Fuchs's test, found by comparing the log likelihood ratio statistic to the chi-squared distribution with the appropriate number of degrees of freedom. Described in Fuchs (1982).

**References**

Fuchs C (1982). "Maximum likelihood estimation and model selection in contingency tables with missing data." *J. Amer. Statist. Assoc.*, **77**(378), 270–278.

**Examples**

```

bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3) # Our canonical 3d example
M=c(2,2,2)
n0=200
nS=c(200,200,200)

pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
P12=pS[1:4]; P13=pS[5:8]; P23=pS[9:12]
X12=t(rmultinom(1,size=nS[1],prob=P12)/nS[1])
X13=t(rmultinom(1,size=nS[2],prob=P13)/nS[2])
X23=t(rmultinom(1,size=nS[3],prob=P23)/nS[3])
pSh=cbind(X12,X13,X23)

p0=array(0.125,dim=c(2,2,2))
p0h=array(rmultinom(1,n0,p0),dim=M)/n0

FuchsTest(p0h,n0,pSh,nS,bS,M,50)

```

---

InconsMinkSumHrep	<i>Calculate the H-representation of the general (possibly inconsistent) Minkowski sum</i>
-------------------	--

---

### Description

Computes the minimal halfspace representation of the Minkowski sum of the marginal polytope and the inconsistent ball defined in Berrett and Samworth (2022).

### Usage

```
InconsMinkSumHrep(bS, M, round = FALSE)
```

### Arguments

bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
round	A logical value indicating whether or not to round coefficients to 15 significant figures. The function RoundErrors can be used separately to substitute other values for 15. Defaults to FALSE.

### Value

A halfspace representation object as used by the rcdd package. See Geyer and Meeden (2021) for more detail.

### References

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.

Geyer CJ, Meeden GD (2021). *rcdd: Computational Geometry*. <https://CRAN.R-project.org/package=rcdd>.

### Examples

```
bS=matrix(c(1,1, 1,0),byrow=TRUE,ncol=2)
InconsMinkSumHrep(bS,c(2,2))
```

---

loglik0	<i>Compute the log likelihood of a probability mass function, under MCAR, given complete and incomplete data</i>
---------	--

---

**Description**

Compute the log likelihood of a probability mass function, under MCAR, given complete and incomplete data

**Usage**

```
loglik0(p, p0h, n0, pSh, nS, bS, M)
```

**Arguments**

p	A probability mass function whose log likelihood is to be calculated.
p0h	An empirical mass function calculated using complete observations.
n0	An integer giving the number of complete observations used to calculate p0h.
pSh	A sequence of empirical mass functions calculated using incomplete observations.
nS	A sequence of integers giving the numbers of incomplete observations used to calculate pSh.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.

**Value**

The value of the log likelihood.

**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3) # Our canonical 3d example
M=c(2,2,2)
n0=200
nS=c(200,200,200)

pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
P12=pS[1:4]; P13=pS[5:8]; P23=pS[9:12]
X12=t(rmultinom(1,size=nS[1],prob=P12)/nS[1])
X13=t(rmultinom(1,size=nS[2],prob=P13)/nS[2])
X23=t(rmultinom(1,size=nS[3],prob=P23)/nS[3])
pSh=cbind(X12,X13,X23)

p0=array(0.125,dim=c(2,2,2))
p0h=array(rmultinom(1,n0,p0),dim=M)/n0
```

```
loglik0(p0,p0h,n0,pSh,nS,bS,M)
```

---

loglik1	<i>Compute the log likelihood of a probability mass function, without assuming MCAR, given complete and incomplete data</i>
---------	---

---

### Description

Compute the log likelihood of a probability mass function, without assuming MCAR, given complete and incomplete data

### Usage

```
loglik1(p0, pS, p0h, n0, pSh, nS, bS, M)
```

### Arguments

p0	A probability mass function on the joint space.
pS	A sequence of probability mass functions on the marginal spaces.
p0h	An empirical mass function calculated using complete observations.
n0	An integer giving the number of complete observations used to calculate p0h.
pSh	A sequence of empirical mass functions calculated using incomplete observations.
nS	A sequence of integers giving the numbers of incomplete observations used to calculate pSh.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.

### Value

The value of the log likelihood.

### Examples

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3) # Our canonical 3d example
M=c(2,2,2)
n0=200
nS=c(200,200,200)

pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
P12=pS[1:4]; P13=pS[5:8]; P23=pS[9:12]
X12=t(rmultinom(1,size=nS[1],prob=P12)/nS[1])
X13=t(rmultinom(1,size=nS[2],prob=P13)/nS[2])
X23=t(rmultinom(1,size=nS[3],prob=P23)/nS[3])
```

```

pSh=cbind(X12,X13,X23)

p0=array(0.125,dim=c(2,2,2))
p0h=array(rmultinom(1,n0,p0),dim=M)/n0

loglik1(p0,pS,p0h,n0,pSh,nS,bS,M)

```

---

MargPolyHrep

*Calculate the H-representation of the marginal polytope*


---

### Description

Computes the minimal halfspace representation of the marginal polytope defined, for example, in Berrett and Samworth (2022).

### Usage

```
MargPolyHrep(bS, M, round = FALSE)
```

### Arguments

bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
round	A logical value indicating whether or not to round coefficients to 15 significant figures. The function RoundErrors can be used separately to substitute other values for 15. Defaults to FALSE.

### Value

A halfspace representation object as used by the rcdd package. See Geyer and Meeden (2021) for more detail.

### References

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.

Geyer CJ, Meeden GD (2021). *rcdd: Computational Geometry*. <https://CRAN.R-project.org/package=rcdd>.

### Examples

```

bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
MargPolyHrep(bS,c(2,2,2))

```

---

MLE	<i>Compute the MLE under MCAR in a contingency table using the EM algorithm, given complete and incomplete observations.</i>
-----	--

---

**Description**

Compute the MLE under MCAR in a contingency table using the EM algorithm, given complete and incomplete observations.

**Usage**

```
MLE(p0h, n0, pSh, nS, bS, M, Niter, loglik = FALSE)
```

**Arguments**

p0h	An empirical mass function calculated using complete observations.
n0	An integer giving the number of complete observations used to calculate p0h.
pSh	A sequence of empirical mass functions calculated using incomplete observations.
nS	A sequence of integers giving the numbers of incomplete observations used to calculate pSh.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
Niter	An integer giving the number of iterations to be used in the EM algorithm.
loglik	A logical value indicating whether or not the log likelihoods at each step of the EM algorithm should be an output. Defaults to FALSE.

**Value**

The output of the EM algorithm, approximating the MLE for the probability mass function on the joint space.

**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3) # Our canonical 3d example
M=c(2,2,2)
n0=200
nS=c(200,200,200)

pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
P12=pS[1:4]; P13=pS[5:8]; P23=pS[9:12]
X12=t(rmultinom(1,size=nS[1],prob=P12)/nS[1])
X13=t(rmultinom(1,size=nS[2],prob=P13)/nS[2])
X23=t(rmultinom(1,size=nS[3],prob=P23)/nS[3])
pSh=cbind(X12,X13,X23)
```

```

p0=array(0.125,dim=c(2,2,2))
p0h=array(rmultinom(1,n0,p0),dim=M)/n0

MLE(p0h,n0,pSh,nS,bS,M,50)

trace=MLE(p0h,n0,pSh,nS,bS,M,50,loglik=TRUE)[[2]]
plot(1:50,trace,type="l")

```

---

ProjectionTest	<i>Carry out a test of MCAR in a contingency table, given incomplete observations.</i>
----------------	--

---

### Description

Carry out a test of MCAR in a contingency table, given incomplete observations.

### Usage

```
ProjectionTest(pSh, nS, bS, M, B)
```

### Arguments

pSh	A sequence of empirical mass functions calculated using incomplete observations.
nS	A sequence of integers giving the numbers of incomplete observations used to calculate pSh.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
B	An integer giving the number of bootstrap samples to be used to calibrate the test.

### Value

The p-value the Monte Carlo test described in Berrett and Samworth (2022).

The value of the test statistic  $R()$ .

### References

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.

**Examples**

```

bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3) # Our canonical 3d example
M=c(2,2,2)
nS=c(200,200,200)

pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
P12=pS[1:4]; P13=pS[5:8]; P23=pS[9:12]
X12=t(rmultinom(1,size=nS[1],prob=P12)/nS[1])
X13=t(rmultinom(1,size=nS[2],prob=P13)/nS[2])
X23=t(rmultinom(1,size=nS[3],prob=P23)/nS[3])
pSh=cbind(X12,X13,X23)

ProjectionTest(pSh,nS,bS,M,99)

```

---

Rindex

*A function computing the incompatibility index*


---

**Description**

A function solving a linear program to compute the incompatibility index  $R()$  defined in Berrett and Samworth (2022), in the case of having discrete random variables. Uses `Amatrix` to define to constraint matrix and `lpSolve` to implement the linear optimisation.

**Usage**

```
Rindex(pS, bS, M)
```

**Arguments**

pS	A sequence of probability mass functions on the marginal spaces.
bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.

**Value**

The value of  $R()$ , in the interval  $[0, 1]$ .

**References**

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.



**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
M=c(2,2,2)
```

```
pS=rep(0.25,12)
Rindex(pS,bS,M)
```

```
pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
Rindex(pS,bS,M)
```

---

RindexDual	<i>A function computing the incompatibility index and associated closest joint mass function using the dual formulation</i>
------------	---

---

**Description**

A function solving a linear program to compute the incompatibility index  $R()$  defined in Berrett and Samworth (2022), in the case of having discrete random variables. Uses `Amatrix` to define to constraint matrix and `lpSolve` to implement the linear optimisation.

**Usage**

```
RindexDual(pS, bS, M, lp_solver = "default", simplex_strategy = 4)
```

**Arguments**

<code>pS</code>	A sequence of probability mass functions on the marginal spaces.
<code>bS</code>	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
<code>M</code>	A vector of positive integers giving the alphabet sizes of the discrete variables.
<code>lp_solver</code>	An argument passed to <code>HiGHS</code> specifying which solver to use. See Schwendinger et al. (2023) for more detail.
<code>simplex_strategy</code>	An argument passed to <code>HiGHS</code> specifying which solver to use. See Schwendinger et al. (2023) for more detail.

**Value**

The value of  $R()$ , in the interval  $[0, 1]$ .  
The optimal solution to the linear program

**References**

Berrett TB, Samworth RJ (2022). “Optimal nonparametric testing of Missing Completely At Random, and its connections to compatibility.” *Preprint, arXiv:2205.08627*.  
Schwendinger F, Schumacher D, Hall J, Galabova I, Gottwald L, Feldmeier M (2023). ‘*HiGHS*’ *Optimization Solver*. <https://CRAN.R-project.org/package=highs>.

**Examples**

```

bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
M=c(2,2,2)
A=Amatrix(bS,M)

pS=rep(0.25,12)
linprog=RindexDual(pS,bS,M)
rbind(pS,as.vector(A%%linprog[[2]]/(1-linprog[[1]]))

pS=c(0.125,0.375,0.375,0.125,0.250,0.250,0.250,0.250,0.100,0.400,0.400,0.100)
linprog=RindexDual(pS,bS,M)
rbind(pS,as.vector(A%%linprog[[2]]/(1-linprog[[1]]))

```

---

RoundErrors

*Round errors in halfspace representations*


---

**Description**

Round errors in halfspace representations

**Usage**

```
RoundErrors(X, digits = 15)
```

**Arguments**

`X`                    A halfspace representation to be rounded.  
`digits`                An integer giving the number of significant figures to be kept.

**Value**

A rounded halfspace representation.

**Examples**

```

bS=matrix(c(1,1,1,0, 1,0,0,1, 0,1,0,1, 0,0,1,1),byrow=TRUE,ncol=4)
RoundErrors("9007199254740992/6004799503160661") #Occurs in ConsMinkSumHrep(bS,c(2,2,2,2))

```

---

row_index	<i>A function indexing the rows of A</i>
-----------	--

---

**Description**

A map from the observation space to an index set.

**Usage**

```
row_index(bS, M, S, xS)
```

**Arguments**

bS	A binary matrix specifying the set of observation patterns. Each row encodes a single pattern.
M	A vector of positive integers giving the alphabet sizes of the discrete variables.
S	An integer indicating which observation pattern is of interest.
xS	An element of the observation space of the specified observation pattern.

**Value**

A positive integer no larger than the cardinality of the joint space uniquely identifying x.

**Examples**

```
bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
M=c(2,2,2)
row_index(bS,M,1,c(1,1))
row_index(bS,M,2,c(1,1))
row_index(bS,M,3,c(1,1))

bS=matrix(c(1,1,0, 1,0,1, 0,1,1),byrow=TRUE,ncol=3)
M=c(4,3,2)
row_index(bS,M,1,c(1,1))
row_index(bS,M,1,c(2,1))
row_index(bS,M,1,c(3,1))
row_index(bS,M,1,c(4,1))
row_index(bS,M,1,c(1,2))
row_index(bS,M,1,c(2,2))
```

# Index

Amatrix, [2](#)  
AmatrixSparse, [3](#)  
  
Cimproved, [3](#)  
col\_index, [4](#)  
ConsMinkSumHrep, [5](#)  
Csimple, [6](#)  
  
EMiteration, [7](#)  
EquivalenceClass, [8](#)  
  
FuchsTest, [8](#)  
  
InconsMinkSumHrep, [10](#)  
  
loglik0, [11](#)  
loglik1, [12](#)  
  
MargPolyHrep, [13](#)  
MLE, [14](#)  
  
ProjectionTest, [15](#)  
  
Rindex, [16](#)  
RindexDual, [17](#)  
RoundErrors, [18](#)  
row\_index, [19](#)