Package: IndependenceTests (via r-universe)

October 25, 2026

Type Package

Title Non-Parametric Tests of Independence Between Random Vectors

Version 0.8

Date 2026-07-16

Maintainer Pierre Lafaye De Micheaux <lafaye@unsw.edu.au>

Description Non-parametric tests of independence (mutual or serial) between some quantitative random vectors, as described in Bilodeau M. and Lafaye de Micheaux P. (2009) <doi:10.1016/j.jspi.2008.11.006>, in Beran R., Bilodeau M. and Lafaye de Micheaux P. (2007) <doi:10.1016/j.jmva.2007.01.009> and in Fan Y., Lafaye de Micheaux P., Penev S. and Salopek D. (2017) <doi:10.1016/j.jmva.2016.09.014>.

License GPL (>= 2)

Depends R (>= 2.6.0), xtable, CompQuadForm, MASS, Runuran, parallel

Copyright For LAPACK/BLAS routines in src (Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd., Courant Institute, Argonne National Lab, and Rice University).

NeedsCompilation yes

Author Pierre Lafaye De Micheaux [aut, cre] (ORCID: <https://orcid.org/0000-0002-0247-5136>), Martin Bilodeau [aut], Yanan Fan [aut], Spiridon Penev [aut], Donna Salopek [aut], Cleve Moler [cph] (LAPACK/BLAS routines in src), Jack Dongarra [cph] (LAPACK/BLAS routines in src), Richard Hanson [cph] (LAPACK/BLAS routines in src), Sven Hammarling [cph] (LAPACK/BLAS routines in src), Jeremy Du Croz [cph] (LAPACK/BLAS routines in src)

Repository https://cran.r-universe.dev

Date/Publication 2026-07-17 06:30:02 UTC

RemoteUrl https://github.com/cran/IndependenceTests

RemoteRef HEAD

RemoteSha 2a724489407aeec75b2cb1b325e8aa49537a0e96

Contents

A.dep.tests 2
dependogram 4
dna 7
highschool 8
mdcov 9

Index **12**

`A.dep.tests` *Tests for mutual or serial independence between categorical variables*

Description

The tests are constructed from the Möbius transformation applied to the probability cells in a multi-way contingency table. The Pearson chi-squared test of mutual independence is partitioned into A-dependence statistics over all subsets A of variables. The goal of the partition is to identify subsets of dependent variables when the mutual independence hypothesis is rejected by the Pearson chi-squared test. The methodology can be directly adapted to test for serial independence of d successive observations of a stationary categorical time series.

For categorical time series, especially those of a nominal (non ordinal) nature, the user should be aware that tests of serial independence obtained by methods suited to quantitative sequences by quantification of the labels are not invariant to permutation of the labels contrary to the test described here.

Usage

```
A.dep.tests(Xmat, choice = 1, d = 0, m = d, freqname = "", type = "text")
```

Arguments

`Xmat`	Table, matrix or data-frame of the contingency table data, if `choice` = 1. Vector of the time series data, if `choice` = 2.
`choice`	Integer. 1 for mutual independence, 2 for serial independence.
`d`	Integer. Used only if `choice` = 2 for the number of successive obervations.
`m`	Integer. Maximum cardinality of subsets A for which an A-dependence statistic is required. This option is particularly useful for large values of `d`.
`freqname`	Character. Used only if `choice` = 1 and when `Xmat` is a matrix or a data-frame to identify the variable for the counts (frequencies).
`type`	"text" or "html"

Value

Returns an object of class `A.dep.tests`, inheriting from `list`, with a `print` method. The object contains the following components:

`TA`	A-dependence statistics for each subset A of variables.
`fA`	degrees of freedom of the A-dependence statistics.
`pvalA`	p-values of the A-dependence statistics.
`X`	summary of the results.
`X2`	test statistic for mutual independence obtained by the sum of the A-dependence statistics, if `choice` = 1.
`Y2`	test statistic for serial independence obtained by the sum of the A-dependence statistics, if `choice` = 2.
`f`	number of degrees of freedom associated with the test statistic `X2` or `Y2`.
`pval`	the p-value associated with the test statistic `X2` or `Y2`.

Author(s)

Bilodeau M., Lafaye de Micheaux P.

References

Bilodeau M., Lafaye de Micheaux P. (2009). A-dependence statistics for mutual and serial independence of categorical variables, Journal of Statistical Planning and Inference, 139, 2407-2419.

Agresti A. (2002). Categorical data analysis, Wiley, p. 322

Whisenant E.C., Rasheed B.K.A., Ostrer H., Bhatnagar Y.M. (1991). Evolution and sequence analysis of a human Y-chromosomal DNA fragment, J. Mol. Evol., 33, 133-141.

Examples

```
# Test of  mutual independence between 3 independent Bernoulli variables.

n <- 100
data <- data.frame(X1 = rbinom(n, 1, 0.3), X2 = rbinom(n, 1, 0.3) , X3 =
                   rbinom(n, 1, 0.3))
X <- table(data)
A.dep.tests(X)

# Test of mutual independence between 4 variables which are
# 2-independent and 3-independent, but are 4-dependent.

n <- 100
W <- sample(x = 1:8, size = n, TRUE)
X1 <- W %in% c(1, 2, 3, 5)
X2 <- W %in% c(1, 2, 4, 6)
X3 <- W %in% c(1, 3, 4, 7)
X4 <- W %in% c(2, 3, 4, 8)
data <- data.frame(X1, X2, X3, X4)
X <- table(data)
```

```
A.dep.tests(X)

# Test of serial independence of a nucleotide sequence of length
# 4156 described in Whisenant et al. (1991).

data(dna)
x2 <- dna[1]
for (i in 2:length(dna)) x2 <- paste(x2, dna[i], sep = "")
x <- unlist(strsplit(x2, ""))
x[x == "a" | x == "g"] <- "r"
x[x == "c" |  x== "t"] <- "y"

out <- A.dep.tests(x, choice = 2, d = 1501, m = 2)$TA[[1]]
plot(100:1500, out[100:1500], xlab = "lag j", ylab = "T(1,j+1)", pch = 19)
abline(h = qchisq(.995, df = 1))


# Analysis of a contingency table in Agresti (2002) p. 322

data(highschool)
A.dep.tests(highschool, freqname = "count")
```

---

dependogram *Nonparametric tests of independence between random vectors*

Description

This function can be used for the following two problems: 1) testing mutual independence between some numerical random vectors, and 2) testing for serial independence of a multivariate stationary quantitative time series. The proposed test does not assume continuous marginals. It is valid for any probability distribution. It is also invariant with respect to the affine general linear group of transformations on the vectors. This test is based on a characterization of mutual independence defined from probabilities of half-spaces in a combinatorial formula of Möbius. As such, it is a natural generalization of tests of independence between univariate random variables using the empirical distribution function. Without the assumption that each vector is one-dimensional with a continuous cumulative distribution function, any test of independence can not be distribution free. The critical values of the proposed test are thus computed with the bootstrap which was shown to be consistent in this context.

Usage

```
dependogram(X, vecd.or.p, N = 10, B = 2000, alpha = 0.05, display =
            TRUE, graphics = TRUE, nbclus = 1)
```

Arguments

X	Data.frame or matrix with observations corresponding to rows and variables to columns.

vecd.or.p	For the mutual independence problem 1), a vector giving the sizes of each subvector. For the serial independence problem 2), an integer indicating the number of consecutive observations.
N	Integer. Number of points of the discretization to obtain directions on the sphere in order to evaluate the value of the test statistic.
B	Integer. Number of bootstrap samples. Note that B can be slightly modified if nbclus > 1
alpha	Double. Global significance level of the test.
display	Logical. TRUE to display values of the A-dependence statistics.
graphics	Logical. TRUE to plot the dependogram.
nbclus	Integer. Number of nodes in the cluster. Used only for parallel computations.

Value

A list with the following components:

In the mutual independence case:

norm.RnA	Supremum norm (Kolmogorov). Test statistic is $\|R_{n,A}\|$ and is computed from the Möbius independence half space processes $R_{n,A}$.
Rn	Maximum value of norm.RnA over all subsets A of variables.
rA	Critical value of the bootstrap distribution of the test statistic $\|R_{n,A}\|$.
r	Critical value of the bootstrap distribution of the test statistic R_n.
RnAsstar	Matrix of size $(2^p - p - 1) \times B$ which contains, for each of the B bootstrap samples, the statistics norm.RnA for all subsets A of variables.

In the serial case:

norm.SnA	Supremum norm (Kolmogorov). Test statistic is $\|S_{n,A}\|$ and is computed from the Möbius independence half space processes $S_{n,A}$.
Sn	Maximum value of norm.SnA over all subsets A of variables.
sA	Critical value of the bootstrap distribution of the test statistic $\|S_{n,A}\|$.
s	Critical value of the bootstrap distribution of the test statistic S_n.
SnAsstar	Matrix of size $(2^{p-1} - 1) \times B$ which contains, for each of the B bootstrap samples, the statistics norm.SnA for all subsets A of variables.

Author(s)

Bilodeau M., Lafaye de Micheaux P.

References

Beran R., Bilodeau M., Lafaye de Micheaux P. (2007). Nonparametric tests of independence between random vectors, Journal of Multivariate Analysis, 98, 1805-1824.

Examples

```
# NOTE: In real applications, B should be set to at least 1000.

# Example 4.1: Test of mutual independence between four discrete Poisson
# variables. The pair (X1,X2) is independent of the pair (X3,X4), with
# each pair having a correlation of 3/4.
# NOTE: with B=1000, this one took 65s with nbclus=1 and 15s with nbclus=7 on my computer.
n <- 100
W1 <- rpois(n, 1)
W3 <- rpois(n, 1)
W4 <- rpois(n, 1)
W6 <- rpois(n, 1)
W2 <- rpois(n, 3)
W5 <- rpois(n, 3)
X1 <- W1 + W2
X2 <- W2 + W3
X3 <- W4 + W5
X4 <- W5 + W6
X <- cbind(X1, X2, X3, X4)
dependogram(X, vecd.or.p = c(1, 1, 1, 1), N = 10, B = 20, alpha = 0.05,
            display = TRUE, graphics = TRUE)

# Example 4.2: Test of mutual independence between three bivariate
#  vectors. The block-diagonal structure of the covariance matrix is
#  such that only the second and third subvectors are dependent.
# NOTE: with B=2000, this one took 3.8h with nbclus=1 on my computer.
n <- 50
mu <- rep(0,6)
Psi <- matrix(c(1, 0, 0, 0, 0, 0,
                0, 1, 0, 0, 0, 0,
                0, 0, 1, 0,.4,.5,
                0, 0, 0, 1,.1,.2,
                0, 0,.4,.1, 1, 0,
                0, 0,.5,.2, 0, 1), nrow = 6, byrow = TRUE)
X <- mvrnorm(n, mu, Psi)

dependogram(X, vecd.or.p = c(2, 2, 2), N = 10, B = 20, alpha = 0.05,
            display = TRUE, graphics = TRUE)


# Example 4.3: Test of mutual independence between 4 dependent binary
# variables which are 2-independent (pairwise) and also 3-independent
# (any 3 of the 4 variables are mutually independent).
n <- 100
W <- sample(x = 1:8, size = n, TRUE)
X1 <- W %in% c(1, 2, 3, 5)
X2 <- W %in% c(1, 2, 4, 6)
X3 <- W %in% c(1, 3, 4, 7)
X4 <- W %in% c(2, 3, 4, 8)
X <- cbind(X1, X2, X3, X4)
dependogram(X, vecd.or.p = c(1, 1, 1, 1), N = 10, B = 20, alpha = 0.05,
            display = TRUE, graphics = TRUE)
```

```
# Example 4.4: Test of serial independence of binary sequences of zeros
# and ones. The sequence W is an i.i.d. sequence. The sequence Y is
# dependent at lag 3.
n <- 100 ; lag <- 3
W <- rbinom(n, 1, 0.8)
Y <- W[1:(n - lag)] * W[(1 + lag):n]
dependogram(W, vecd.or.p = 4, N = 10, B = 20, alpha = 0.05, display =
            TRUE, graphics = TRUE)
dependogram(Y, vecd.or.p = 4, N = 10, B = 20, alpha = 0.05, display =
            TRUE, graphics = TRUE)

# Example 4.5: Test of serial independence of sequences of directional
# data on the 2-dimensional sphere. The sequence W is an
# i.i.d. sequence. The sequence Y is dependent at lag 1.
# NOTE: with B=2000, this one took 7.9h with nbclus=1 on my computer.
n <- 75 ; lag <- 1
U <- matrix(rnorm(2 * n), nrow = n, ncol = 2)
W <- U[1:(n - lag),] + sqrt(2) * U[(1 + lag):n,]
Y <- W / apply(W, MARGIN = 1, FUN = function(x) {sqrt(x[1] ^ 2 + x[2] ^ 2)})

dependogram(Y, vecd.or.p = 3, N = 10, B = 20, alpha = 0.05, display =
            TRUE, graphics = TRUE)


# This one always gives the same value of the test statistic:
x <- rnorm(100)
dependogram(X = cbind(x, x), vecd.or.p = c(1, 1), N = 2, B = 2, alpha =
0.05, display = FALSE, graphics = FALSE, nbclus = 1)$Rn
# This is correct because this is equivalent to computing:
I <- 1:100
n <- 100
sqrt(n) * max(I/n - I ^ 2 / n ^ 2)
```

---

dna *dna sequence*

Description

This data from Whisenant et al. (1991) is a nucleotides sequence of 4156 base pairs (bp). The categorical variable represents the nucleotide which is either one of the two purines (r), adenine (a) or guanine (g), or one of the two pyrimidines (y), cytosine (c) or thymine (t).

Usage

```
data(dna)
```

Format

A character vector of length 70 representing 70 consecutive segments of a dna strand of length 4156.

References

Whisenant E.C., Rasheed B.K.A., Ostrer H., Bhatnagar Y.M. (1991). Evolution and sequence analysis of a human Y-chromosomal DNA fragment, J. Mol. Evol., 33, 133-141.

Examples

```
data(dna)
x2 <- dna[1]
for (i in 2:length(dna)) x2 <- paste(x2, dna[i], sep = "")
x <- unlist(strsplit(x2, ""))
```

---

highschool — *Highschool data on alcohol, cigarette and marijuana use for high-school seniors*

Description

Data from a 1992 survey by the Wright State University School of Medecine and the United Health Services in Dayton, Ohio. The survey asked 2276 students in their final year of highschool in a nonurban area near Dayton, Ohio, whether they had ever used alcohol, cigarettes, or marijuana.

Usage

```
data(highschool)
```

Format

A data frame of 8 observations on the variables:

alcohol a factor vector with components "yes" or "no".

cigarette a factor vector with components "yes" or "no".

marijuana a factor vector with components "yes" or "no".

count a numeric vector of frequencies.

References

Agresti A. (2002). Categorical data analysis, Wiley, p. 322.

Examples

```
data(highschool)
```

mdcov *Computation of the multidimensional distance covariance statistic for mutual independence using characteristic functions.*

Description

Computation of the multidimensional distance covariance statistic for mutual independence using characteristic functions. Compute the eigenvalues associated with the empirical covariance of the limiting Gaussian procces. Compute the p-value associated with the test statistic, using the Imhof procedure.

Usage

```
mdcov(X, vecd, a = 1, weight.choice = 1, N = 200, cubature = FALSE, K =
100, epsrel = 10 ^ -6, norming = TRUE, thresh.eigen = 10 ^ -8, estim.a =
FALSE, Cpp = TRUE, pval.comp = TRUE)
```

Arguments

X	Data.frame or matrix with observations corresponding to rows and variables to columns.
vecd	a vector giving the sizes of each subvector.
a	parameter for the weight function.
weight.choice	Integer value in 1, 2, 3, 4, 5 corresponding to the choice in our paper.
N	Number of Monte-Carlo samples.
cubature	Logical. If FALSE, a Monte-Carlo approach is used. If TRUE, a cubature approach is used.
K	Number of eigenvalues to compute.
epsrel	relative accuracy requested for the Imhof procedure.
norming	Logical. Should we normalize the test statistic with H_n.
thresh.eigen	We will not compute eigenvalues (involved in the limiting distribution) below that threshold.
estim.a	Logical. Should we automatically estimate the value of a.
Cpp	Logical. If TRUE computations will be done using a fast C code. The use of FALSE is only useful to compare the results with the one given by the C code.
pval.comp	Logical. If FALSE do not compute the p-values and lambdas.

Value

A list with the following components:

mdcov	the value of the statistic $nT_n(w)$ (this value has been normed if norming = TRUE)
Hn	the denominator of $nT_n(w)$, namely H_n
pvalue	the p-value of the test
lambdas	the vector of eigenvalues computed (they have not been divided by their sum)

Author(s)

Lafaye de Micheaux P.

References

Fan Y., Lafaye de Micheaux P., Penev S. and Salopek D. (2017). Multivariate nonparametric test of independence, Journal of Multivariate Analysis, 153, 189-210.

Examples

```
a <- 1

# 4.1 Dependence among four discrete variables
set.seed(1)
n <- 100
w1 <- rpois(n, 1)
w3 <- rpois(n, 1)
w4 <- rpois(n, 1)
w6 <- rpois(n, 1)
w2 <- rpois(n, 3)
w5 <- rpois(n, 3)
x1 <- w1 + w2
x2 <- w2 + w3
x3 <- w4 + w5
x4 <- w5 + w6

X <- cbind(x1, x2, x3, x4)
mdcov(X, vecd = c(2, 2), a, weight.choice = 1, N = 100, cubature = TRUE)
mdcov(X, vecd = c(1, 1, 1, 1), a, weight.choice = 1, N = 100, cubature = TRUE)

X <- cbind(x1, x2)
mdcov(X, vecd = c(1, 1), a, weight.choice = 1, N = 100, cubature = TRUE)

X <- cbind(x3, x4)
mdcov(X, vecd = c(1, 1), a, weight.choice = 1, N = 100, cubature = TRUE)

# 4.2 Dependence between three bivariate vectors
set.seed(2)
n <- 200
Sigma <- matrix(c(
1 , 0 ,  0 ,  0 ,  0 ,  0 ,
0 , 1 ,  0 ,  0 ,  0 ,  0 ,
0 , 0 ,  1 ,  0 , .4 , .5 ,
0 , 0 ,  0 ,  1 , .1 , .2 ,
0 , 0 , .4 , .1 ,  1 ,  0 ,
0 , 0 , .5 , .2 ,  0 ,  1 ) ,
nrow = 6 , ncol = 6)
W <- mvrnorm(n = n, mu = rep(0,6), Sigma = Sigma, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
mdcov(W, vecd = c(2, 2, 2), a, weight.choice = 1, N = 100, cubature = TRUE, epsrel = 10 ^ -7)

 # X^{(1)} with X^{(2)}^2
mdcov(W[,1:4], vecd = c(2, 2), a, weight.choice = 1, N = 100, cubature = TRUE)
```

```
# X^{(2)} with X^{(3)}^2
mdcov(W[,2:6], vecd = c(2, 2), a, weight.choice = 1, N = 100, cubature = TRUE)

# X^{(1)} with X^{(3)}^2
mdcov(W[,c(1:2, 4:6)], vecd = c(2, 2), a, weight.choice = 1, N = 100, cubature = TRUE)

# 4.3 Four-dependent variables which are 2-independent and 3-independent
set.seed(3)
n <- 300
W <- sample(1:8, n, replace = TRUE)
X1 <- W %in% c(1, 2, 3, 5)
X2 <- W %in% c(1, 2, 4, 6)
X3 <- W %in% c(1, 3, 4, 7)
X4 <- W %in% c(2, 3, 4, 8)
X <- cbind(X1, X2, X3, X4)
   # pairwise independence
mdcov(X[,c(1, 2)], vecd = c(1, 1), a, weight.choice = 1, cubature = TRUE)
mdcov(X[,c(1, 3)], vecd = c(1, 1), a, weight.choice = 1, N = 100, cubature = TRUE)
mdcov(X[,c(1, 4)], vecd = c(1, 1), a, weight.choice = 1, cubature = TRUE)
mdcov(X[,c(2, 3)], vecd = c(1, 1), a, weight.choice = 1, cubature = TRUE)
mdcov(X[,c(2, 4)], vecd = c(1, 1), a, weight.choice = 1, cubature = TRUE)
mdcov(X[,c(3, 4)], vecd = c(1, 1), a, weight.choice = 1, cubature = TRUE)
   # 3-independence
mdcov(X[,c(1, 2, 3)], vecd = c(1, 1, 1), a, weight.choice =
                  1, cubature = TRUE)
mdcov(X[,c(1, 2, 4)], vecd = c(1, 1, 1), a, weight.choice =
                  1, cubature = TRUE)
mdcov(X[,c(1, 3, 4)], vecd = c(1, 1, 1), a, weight.choice =
                  1, cubature = TRUE)
mdcov(X[,c(2, 3, 4)], vecd = c(1, 1, 1), a, weight.choice =
                  1, cubature = TRUE)
   # 4-dependence
mdcov(X, vecd = c(1, 1, 1, 1), a, weight.choice = 1, cubature = TRUE)
```

Index

∗ **datasets**
dna, 7
highschool, 8

A.dep.tests, 2

dependogram, 4
dna, 7

highschool, 8

mdcov, 9