

Package ‘discnorm’

May 8, 2026

Type Package

Title Test for Discretized Normality in Ordinal Data

Version 0.2.1

Description Tests whether multivariate ordinal data may stem from discretizing a multivariate normal distribution.
The test is described by Foldnes and Grønneberg (2019) <[doi:10.1080/10705511.2019.1673168](https://doi.org/10.1080/10705511.2019.1673168)>. In addition, an adjusted polychoric correlation estimator is provided that takes marginal knowledge into account, as described by Grønneberg and Foldnes (2022) <[doi:10.1037/met0000495](https://doi.org/10.1037/met0000495)>.

License GPL (>= 2)

Encoding UTF-8

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports lavaan (>= 0.6.10), arules, sirt, MASS, pbivnorm, cubature, copula, mnormt, GoFKernel

RoxygenNote 7.1.1

NeedsCompilation no

Author Njål Foldnes [aut, cre],
Steffen Grønneberg [aut]

Maintainer Njål Foldnes <njal.foldnes@gmail.com>

Repository CRAN

Date/Publication 2022-05-25 15:20:02 UTC

Contents

bootTest	2
catLSadj	3

Index	6
--------------	----------

bootTest	<i>Bootstrap test for discretized normality</i>
----------	---

Description

bootTest is a bootstrap test for whether an ordinal dataset is consistent with being a discretization of a multivariate normal dataset.

Usage

```
bootTest(my.data, B = 1000, verbose = TRUE)
```

Arguments

my.data	A dataset containing ordinal data. Must contain only integer values.
B	Number of bootstrap samples.
verbose	If true, bootstrap progress is printed to the console.

Value

p-value associated with the underlying normality hypothesis.

References

Njål Foldnes & Steffen Grønneberg (2019) Pernicious Polychorics: The Impact and Detection of Underlying Non-normality, Structural Equation Modeling: A Multidisciplinary Journal, DOI: 10.1080/10705511.2019.1673168

Examples

```
set.seed(1)
norm.data <- MASS::mvrnorm(300, m=rep(0,3),
Sigma=cov(MASS::mvrnorm(15, mu=rep(0,3), Sigma=diag(3))))
disc.data <- apply(norm.data,2, cut,
breaks = c(-Inf, 0,1, Inf), labels=FALSE)# normal data discretized
pvalue <- bootTest(disc.data, B=500)
#no support for underlying non-normality
```

catLSadj	<i>Adjusted polychoric correlation</i>
----------	--

Description

catLSadj estimates the underlying correlations assuming bivariate normal copulas, and marginal underlying distributions as provided by the user. The output, i.e., an adjusted correlation matrix and its associated asymptotic covariance matrix, may be used to estimate ordinal factor models and structural equation models with systems such as lavaan. How to do this is shown in the example/vignette.

Usage

```
catLSadj(data.df, marginslist, verbose = FALSE)
```

Arguments

data.df	A dataset containing ordinal data.
marginslist	<p>A list of length equal to the number of columns in data.df. Each element in the list specifies a univariate marginal distribution. Each element must contain a function F, which is the univariate CDF of the marginal. It must accept vectorial input.</p> <p>F is assumed to be continuous, and have support on a interval, which may be all numbers. That is, a random variable X which is F distributed can take any values in an interval (which could be the set of all real numbers). In addition, elements "qF", "sd" and "support" may be included. "qF" is to be the quantile function of F. "sd" is to be the standard deviation of F. "support" is the support of the distribution of F. If "support" is not included, it is assumed to be all numbers. If qF or sd is not included, they are numerically approximated.</p> <p>For optimal performance, both qF and sd should be passed. If they are not provided, they will be approximated numerically, sometimes at great cost of both precision and execution speed. For all well-known univariate distributions, both qF and sd are well-known. Implementations for most quantiles are available in R, and standard deviation formulas for most distributions are available on Wikipedia.</p>
verbose	If true, additional information is printed to screen.

Value

A list of two elements: The adjusted polychoric correlation matrix and its associated asymptotic covariance matrix.

References

Steffen Grønneberg & Njål Foldnes (2022) Factor Analyzing Ordinal Items Requires Substantive Knowledge of Response Marginals, Psychological Methods, DOI: 10.1037/met0000495

Examples

```

shape= 2
scale = 1/sqrt(shape)
m1 <- list(F=function(x) pchisq(x, df=1), qF=function(x) qchisq(x, df=1), sd=sqrt(2))
G3 <- function(x) pgamma(x+shape*scale, shape=shape, scale=scale)
G3flip <- function(x) 1- G3(-x)
qG3 <- function(x) qgamma(x, shape=shape, scale=scale)-shape*scale
qG3flip <- function(x) -qG3(1-x)
marginslist <- list(m1, list(F=G3, qF=qG3), list(F=G3flip, qF=qG3flip))
Sigma <- diag(3)
Sigma[Sigma==0] <- 0.5
Sigma
# [,1] [,2] [,3]
# [1,] 1.0 0.5 0.5
# [2,] 0.5 1.0 0.5
# [3,] 0.5 0.5 1.0
set.seed(1)
norm.data <- MASS::mvrnorm(10^5, rep(0,3), Sigma)
colnames(norm.data) <- c("x1", "x2", "x3")
#With normal marginals, the correlation matrix is (approximately)
#Sigma.
#Transform the marginals to follow the elements in marginslist:
nonnorm.data <- data.frame(x1=marginslist[[1]]$qF(pnorm(norm.data[, 1])),
                          x2=marginslist[[2]]$qF(pnorm(norm.data[, 2])),
                          x3=marginslist[[3]]$qF(pnorm(norm.data[, 3])))
#the transformed data has the following correlation
cor(nonnorm.data)
#           x1           x2           x3
# x1 1.0000000 0.4389354 0.3549677
# x2 0.4389354 1.0000000 0.4256547
# x3 0.3549677 0.4256547 1.0000000
#The original normal dataset ia fitted perfectly to a factor model with
#the following parameters:
head(lavaan::standardizedsolution(lavaan::cfa("F=~ x1+x2+x3", norm.data)),3)
#Extract from output:
#   lhs op rhs est.std   se      z pvalue ci.lower ci.upper
# 1  F =~ x1  0.710 0.002 286.201    0   0.705   0.715
# 2  F =~ x2  0.707 0.002 284.859    0   0.702   0.712
# 3  F =~ x3  0.710 0.002 286.078    0   0.705   0.715

#With non-normal marginals, the factor loadings (and remaining parameters)
#change:
head(lavaan::standardizedsolution(lavaan::cfa("F=~ x1+x2+x3", nonnorm.data)),3)
#Extract from output:
#   lhs op rhs est.std   se      z pvalue ci.lower ci.upper
# 1  F =~ x1  0.605 0.003 191.004    0   0.599   0.611
# 2  F =~ x2  0.725 0.003 219.829    0   0.719   0.732
# 3  F =~ x3  0.587 0.003 185.940    0   0.581   0.593
#Discretize the non-normal dataset:
disc.data <- data.frame(x1=cut(nonnorm.data[, 1], breaks= c(-Inf, 0.1, 1, Inf), labels=FALSE),
                       x2= cut(nonnorm.data[, 2], breaks= c(-Inf, -.7, 0,1, Inf), labels=FALSE),
                       x3=cut(nonnorm.data[, 3], breaks= c(-Inf, -1, 0,1, Inf), labels=FALSE))

```

```

#The polychoric correlation is not close to the correlation matrix
#of the non-normal data, but is close to the correlation matrix
#of the -normal- dataset:
lavaan::lavCor(disc.data, ordered=colnames(disc.data))
#   x1   x2   x3
# x1 1.000
# x2 0.503 1.000
# x3 0.506 0.503 1.000
#Compute the adjustments:
## Not run: adjusted <- catLSadj(disc.data, marginslist, verbose=T )

#The estimated adjusted polychoric correlation matrix is close to the
#correlation matrix of the non-normal data:
adjusted[[1]]
#   x1   x2   x3
# x1 1.000
# x2 0.440 1.000
# x3 0.357 0.427 1.000
# run cat LS without adjustment
fcat <- lavaan::cfa("F=~ x1+x2+x3", disc.data, ordered=colnames(disc.data))
head(lavaan::standardizedsolution(fcat), 3)
#Extract from output:
#   lhs op rhs est.std   se      z pvalue ci.lower ci.upper
# 1  F =~ x1  0.712 0.003 221.553    0   0.705   0.718
# 2  F =~ x2  0.707 0.003 225.835    0   0.701   0.713
# 3  F =~ x3  0.711 0.003 226.119    0   0.705   0.717
#These parameter estimates are close to the parameters of the continuous model for
#normal data, and not to the model parameters obtained from the discretized non-normal dataset
#To get consistent estimates of these parameters
#we need to use the adjusted polychoric correlation.
#To get lavaan to compute the adjusted factor estimates, we need:
sample.th <- lavaan::lavInspect(fcat, "sampstat")$th
attr(sample.th, "th.idx") <- lavaan::lavInspect(fcat, "th.idx")
#the asymptotic covariance matrix of the adjusted polychorics:
gamma.adj <- adjusted[[2]]
WLS.V.new <- diag(1/diag(gamma.adj))
fcat.adj <- lavaan::cfa("F=~ x1+x2+x3", sample.cov=adjusted[[1]],
                      sample.nobs=nrow(disc.data), sample.th=sample.th,
                      NACOV = gamma.adj, WLS.V=WLS.V.new)
head(lavaan::standardizedsolution(fcat.adj), 3)
#Extract from output:
#   lhs op rhs est.std   se      z pvalue ci.lower ci.upper
# 1  F =~ x1  0.607 0.003 224.011    0   0.602   0.612
# 2  F =~ x2  0.725 0.003 224.485    0   0.719   0.731
# 3  F =~ x3  0.589 0.002 237.887    0   0.584   0.593
#Closely matches the model parameters obtained with the non-normal dataset
## End(Not run)

```

Index

bootTest, [2](#)

catLSadj, [3](#)