Package ‘CorrToolBox’

March 5, 2021

Type Package

Title Modeling Correlational Magnitude Transformations in Discretization Contexts

Version 1.6.3

Date 2021-03-05

Author Rawan Allozi, Hakan Demirtas, Ran Gao

Maintainer Ran Gao <rgao8@uic.edu>

Description Modeling the correlation transitions under specified distributional assumptions within the realm of discretization in the context of the latency and threshold concepts. The details of the method are explained in Demirtas, H. and Vardar-Acar, C. (2017) <DOI:10.1007/978-981-10-3307-0_4>.

License GPL-2 | GPL-3

Imports BinNonNor, BinOrdNonNor, GenOrd, moments, mvtnorm, psych

NeedsCompilation no

Repository CRAN

Date/Publication 2021-03-05 18:40:05 UTC

R topics documented:

`CorrToolBox-package` *Modeling Correlational Magnitude Transformations in Discretization Contexts*

Description

This package implements the computational algorithms for modeling the correlation transitions under specified distributional assumptions within the realm of discretization in the context of the latency and threshold concepts. Functions that compute the correlational magnitude changes in both directions (identification of the pre-discretization correlation value in order to attain a specified post-discretization magnitude, and the other way around) are provided.

This package consists of eight main functions. Computing the tetrachoric correlation from the phi coefficient and vice versa are done in `phi2tet` and `tet2phi`, respectively. Computing the polychoric correlation from the ordinal phi coefficient and vice versa are done in `ophi2poly` and `poly2ophi`, respectively. Computing the biserial correlation from the point-biserial correlation and vice versa are done in `pbs2bs` and `bs2pbs`, respectively. Computing the polyserial correlation from the point-polyserial correlation and vice versa are done in `pps2ps` and `ps2pps`, respectively.

Auxiliary functions are also provided. `corrY2corrZ`, `corrZ2corrY`, `corrZ2ophi`, `corrZ2phi`, and `ophi2corrZ` are intermediate functions utilized within the main functions but can be used as standalone functions. `ordY` discretizes a continuous variable, and `mps2cps` provides cumulative probabilities for each set of marginal probabilities in a list. Additional intermediate functions from imported packages include `phi2tetra` from the `psych` package, `ordcont` and `contord` from the `GenOrd` package, `skewness` and `kurtosis` from the `moments` package, `validation.skewness.kurtosis` from the `BinNonNor` package, and `pmvnorm` from the `mvtnorm` package.

Within each correlation transition function, the correlation boundaries for the given marginal distributions are compared to the specified input correlation to ensure there are no violations according to Demirtas and Hedeker (2011). The function `valid.limits.BinOrdNN` in the package `BinOrdNonNor` is utilized for this step. Additionally, `Fleishman.coef.NN` in the package `BinOrdNonNor` is used wherever Fleishman coefficients need to be calculated for a continuous variable.

Details

Package:	CorrToolBox
Type:	Package
Version:	1.6.3
Date:	2021-03-05
License:	GPL-2 | GPL-3

Author(s)

Rawan Allozi, Hakan Demirtas, Ran Gao

Maintainer: Ran Gao <rgao8@uic.edu>

References

Demirtas, H. (2016). A note on the relationship between the phi coefficient and the tetrachoric correlation under nonnormal underlying distributions. *The American Statistician*, **70(2)**, 143-148.

Demirtas, H., Ahmadian, R., Atis, S., Can, F.E., and Ercan, I. (2016). A nonnormal look at polychoric correlations: modeling the change in correlations before and after discretization. *Computational Statistics*, **31(4)**, 1385-1401.

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

Demirtas, H. and Hedeker, D. (2016). Computing the point-biserial correlation under any underlying continuous distribution. *Communications in Statistics-Simulation and Computation*, **45(8)**, 2744-2751.

Demirtas, H., Hedeker, D., and Mermelstein, R. J. (2012). Simulation of massive public health data by power polynomials. *Statistics in Medicine*, **31(27)**, 3337-3346.

Demirtas, H. and Vardar-Acar, C. (2017). Anatomy of correlational magnitude transformations in latency and discretization contexts in Monte-Carlo studies. In ICSA Book Series in Statistics, John Dean Chen and Ding-Geng (Din) Chen (Eds): *Monte-Carlo Simulation-Based Statistical Modeling*. Singapore: Springer, 59-84.

Ferrari, P.A. and Barbiero, A. (2012). Simulating ordinal data. *Multivariate Behavioral Research*, **47(4)**, 566-589.

Fleishman A.I. (1978). A method for simulating non-normal distributions. *Psychometrika*, **43(4)**, 521-532.

Vale, C.D. and Maurelli, V.A. (1983). Simulating multivariate nonnormal distributions. *Psychometrika*, **48(3)**, 465-471.

`bs2pbs` *Computation of the Point-Biserial Correlation from the Biserial Correlation*

Description

This function computes the point-biserial correlation between two variables after one of the variables is dichotomized given the correlation before dichotomization (biserial correlation) as seen in Demirtas and Hedeker (2016). Before computation of the point-biserial correlation, the specified biserial correlation is compared to the lower and upper correlation bounds of the two continuous variables using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
bs2pbs(bs, bin.var, cont.var, p=NULL, cutpoint=NULL)
```

Arguments

bs	The biserial correlation.
bin.var	A numeric vector of the continuous variable before dichotomization.
cont.var	A numeric vector of the continuous variable that is not transformed.
p	The expected value of the numeric vector `bin.var` after dichotomization. Either `p` or `cutpoint` should be specified.
cutpoint	The value at which the numeric vector `bin.var` should be dichotomized. Either `p` or `cutpoint` should be specified.

Value

The point-biserial correlation.

References

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

Demirtas, H. and Hedeker, D. (2016). Computing the point-biserial correlation under any underlying continuous distribution. *Communications in Statistics-Simulation and Computation*, **45(8)**, 2744-2751.

Examples

```
set.seed(123)
y1<-rweibull(n=100000, scale=1, shape=1.2)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=3, s2=1, pi=0.6)

bs2pbs(bs=0.6, bin.var=y1, cont.var=y2, p=0.55)
bs2pbs(bs=0.6, bin.var=y1, cont.var=y2, cutpoint=0.65484)
```

---

corrY2corrZ — *Computation of the Correlation of Bivariate Standard Normal Variables from the Correlation of Bivariate Nonnormal Variables*

Description

This is an intermediate function that computes the correlation of bivariate standard normal variables from the correlation of continuous nonnormal variables. Fleishman coefficients for each nonnormal variable with the specified skewness and excess kurtosis are found. The Fleishman coefficients and correlation of nonnormal variables are used to find the correlation of the two respective standard normal variables as seen in Demirtas, Hedeker, and Mermelstein (2012).

Usage

```
corrY2corrZ(corrY, skew.vec, kurto.vec)
```

Arguments

`corrY`	The correlation of two continuous nonnormal variables.
`skew.vec`	The skewness vector for continuous variables.
`kurto.vec`	The kurtosis vector for continuous variables.

Value

The correlation of the two respective standard normal variables.

References

Demirtas, H., Hedeker, D., and Mermelstein, R. J. (2012). Simulation of massive public health data by power polynomials. *Statistics in Medicine*, **31(27)**, 3337-3346.

Fleishman A.I. (1978). A method for simulating non-normal distributions. *Psychometrika*, **43(4)**, 521-532.

See Also

`tet2phi`, `poly2ophi`

Examples

```
set.seed(987)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=1)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=3, s2=1, pi=0.5)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

corrY2corrZ(corrY=-0.4, skew.vec=c(y1.skew, y2.skew), kurto.vec=c(y1.exkurt, y2.exkurt))
```

corrZ2corrY — *Computation of the Correlation of Bivariate Nonnormal Variables from the Correlation of Bivariate Standard Normal Variables*

Description

Fleishman coefficients for each nonnormal continuous variable with the specified skewness and excess kurtosis are found. The Fleishman coefficients and correlation of two standard normal variables are used to find the correlation of the two nonnormal variables as described in Demirtas, Hedeker, and Mermelstein (2012).

Usage

```
corrZ2corrY(corrZ, skew.vec, kurto.vec)
```

Arguments

`corrZ`	The correlation of two standard normal variables.
`skew.vec`	The skewness vector for continuous variables.
`kurto.vec`	The kurtosis vector for continuous variables.

Value

The correlation of two continuous nonnormal variables as defined by the skewness and excess kurtosis vectors.

References

Demirtas, H., Hedeker, D., and Mermelstein, R. J. (2012). Simulation of massive public health data by power polynomials. *Statistics in Medicine*, **31(27)**, 3337-3346.

Fleishman A.I. (1978). A method for simulating non-normal distributions. *Psychometrika*, **43(4)**, 521-532.

See Also

`phi2tet`

Examples

```
set.seed(987)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=1)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
```

```
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=3, s2=1, pi=0.5)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

corrZ2corrY(corrZ=-0.849, skew.vec=c(y1.skew, y2.skew), kurto.vec=c(y1.exkurt, y2.exkurt))
```

---

corrZ2ophi *Computation of the Ordinal Phi Coefficient from the Correlation of Bivariate Standard Normal Variables*

Description

This is an intermediate function that utilizes `mps2cps` to transform the specified marginal probabilities into cumulative probabilities and uses the `contord` function in the `GenOrd` package to compute the ordinal phi coefficient derived from discretizing bivariate standard normal variables.

Usage

```
corrZ2ophi(corrZ, p1, p2)
```

Arguments

`corrZ`	The correlation of two standard normal variables.
`p1`	A numeric vector containing marginal probabilities defining categories for the first ordinal variable.
`p2`	A numeric vector containing marginal probabilities defining categories for the second ordinal variable.

Value

The ordinal phi coefficient.

References

Demirtas, H., Ahmadian, R., Atis, S., Can, F.E., and Ercan, I. (2016). A nonnormal look at polychoric correlations: modeling the change in correlations before and after discretization. *Computational Statistics*, **31(4)**, 1385-1401.

Ferrari, P.A. and Barbiero, A. (2012). Simulating ordinal data. *Multivariate Behavioral Research*, **47(4)**, 566-589.

See Also

`mps2cps`, `poly2ophi`

Examples

```
set.seed(567)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=3.6)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=2, s2=1, pi=0.3)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

corrZ2ophi(corrZ=0.502, p1=c(0.4, 0.3, 0.2, 0.1), p2=c(0.2, 0.2, 0.6))
```

---

corrZ2phi — *Computation of the Phi Coefficient from the Correlation of Bivariate Standard Normal Variables*

Description

This function computes the phi coefficient derived from dichotomizing bivariate standard normal variables.

Usage

```
corrZ2phi(corrZ, p1, p2)
```

Arguments

corrZ	The correlation of two standard normal variables.
p1	The expected value of the first variable after dichotomization.
p2	The expected value of the second variable after dichotomization.

Value

The phi coefficient.

References

Demirtas, H. (2016). A note on the relationship between the phi coefficient and the tetrachoric correlation under nonnormal underlying distributions. *The American Statistician*, **70(2)**, 143-148.

See Also

tet2phi

Examples

```
set.seed(987)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=1)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=3, s2=1, pi=0.5)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

corrZ2phi(corrZ=-0.456, p1=0.85, p2=0.15)
```

---

mps2cps — *Computation of Cumulative Probabilities Given a Set of Marginal Probabilities*

Description

This function computes cumulative probabilities for each ordinal variable as defined by marginal probabilities provided in a list.

Usage

```
mps2cps(mps)
```

Arguments

`mps`	A list of marginal probability vectors corresponding to each ordinal variable. Each vector within the list `mps` must sum to 1.

Value

A list of vectors containing cumulative probabilities for each set of marginal probabilities specified in `mps`. The i-th element of the list is a vector of the cumulative probabilities defining the marginal distribution of the i-th element of `mps`. If the i-th variable has k categories, the i-th vector in the output will contain (k-1) probability values. The k-th element is implicitly 1.

Examples

```
mps2cps(list(c(0.4, 0.3, 0.2, 0.1), c(0.2, 0.2, 0.6)))
```

ophi2corrZ — *Computation of the Correlation of Bivariate Standard Normal Variables from the Ordinal Phi Coefficient*

Description

This is an intermediate function that transforms marginal probabilities into cumulative probabilities and uses the `ordcont` function in the `GenOrd` package to compute the correlation of bivariate standard normal variables from the ordinal phi coefficient.

Usage

```
ophi2corrZ(ophi, p1, p2)
```

Arguments

`ophi`	The ordinal phi coefficient.
`p1`	A numeric vector containing marginal probabilities defining categories for the first ordinal variable.
`p2`	A numeric vector containing marginal probabilities defining categories for the second ordinal variable.

Value

The correlation of standard normal variables.

References

Demirtas, H., Ahmadian, R., Atis, S., Can, F.E., and Ercan, I. (2016). A nonnormal look at polychoric correlations: modeling the change in correlations before and after discretization. *Computational Statistics*, **31(4)**, 1385-1401.

Ferrari, P.A. and Barbiero, A. (2012). Simulating ordinal data. *Multivariate Behavioral Research*, **47(4)**, 566-589.

See Also

`mps2cps`, `ophi2poly`

Examples

```
set.seed(567)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=3.6)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)
```

```
gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=2, s2=1, pi=0.3)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

ophi2corrZ(ophi=-0.7, p1=c(0.4, 0.3, 0.2, 0.1), p2=c(0.2, 0.2, 0.6))
```

---

ophi2poly — *Computation of the Polychoric Correlation from the Ordinal Phi Coefficient*

Description

This function computes the polychoric correlation between two continuous variables given the correlation after ordinalization of both variables (ordinal phi coefficient) as seen in Demirtas et al. (2016). Before computation of the polychoric correlation, the specified ordinal phi coefficient is compared to the lower and upper correlation bounds of the two ordinal variables using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
ophi2poly(ophicoef, dist1, dist2)
```

Arguments

`ophicoef`	The ordinal phi coefficient.
`dist1`	A list of length 3 containing the skewness, excess kurtosis, and a numeric vector of marginal probabilities after dichotomization for the first continuous variable with names skewness, exkurtosis, and p, respectively.
`dist2`	A list of length 3 containing the skewness, excess kurtosis, and a numeric vector of marginal probabilities after dichotomization for the second continuous variable with names skewness, exkurtosis, and p, respectively.

Value

The polychoric correlation.

References

Demirtas, H., Ahmadian, R., Atis, S., Can, F.E., and Ercan, I. (2016). A nonnormal look at polychoric correlations: modeling the change in correlations before and after discretization. *Computational Statistics*, **31(4)**, 1385-1401.

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

Ferrari, P.A. and Barbiero, A. (2012). Simulating ordinal data. *Multivariate Behavioral Research*, **47(4)**, 566-589.

See Also

corrZ2corrY, ophi2corrZ, mps2cps

Examples

```
set.seed(567)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=3.6)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=2, s2=1, pi=0.3)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

ophi2poly(ophicoef=-0.7,
          dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=c(0.4, 0.3, 0.2, 0.1)),
          dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt, p=c(0.2, 0.2, 0.6)))

ophi2poly(ophicoef=0.2,
          dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=c(0.1, 0.1, 0.1, 0.7)),
          dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt, p=c(0.8, 0.1, 0.1)))
```

---

ordY *Ordinalization of a Continuous Variable*

Description

This functions creates an ordinalized form of a continuous variable.

Usage

```
ordY(mp, cat, y)
```

Arguments

mp	A vector of marginal probabilities defining the ordinalized variable.
cat	A numeric vector containing the categories for each respective marginal probability in mp.
y	A continuous variable to be ordinalized into categories in cat as defined by mp.

Value

A data frame containing the given continuous variable and the ordinalized variable with names y and x, respectively.

See Also

mps2cps

Examples

```
y<-rnorm(100000)
dat<-ordY(mp=c(0.25, 0.5, 0.25), cat=c(1,2,3), y=y)
```

---

pbs2bs *Computation of the Biserial Correlation from the Point-Biserial Correlation*

Description

This function computes the biserial correlation between two continuous variables given the correlation after dichotomization of one of the variables (point-biserial correlation) as seen in Demirtas and Hedeker (2016). Before computation of the biserial correlation, the specified point-biserial correlation is compared to the lower and upper correlation bounds of the continuous variable and binary variable using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
pbs2bs(pbs, bin.var, cont.var, p=NULL, cutpoint=NULL)
```

Arguments

`pbs`	The point-biserial correlation.
`bin.var`	A numeric vector of the continuous variable before dichotomization.
`cont.var`	A numeric vector of the the continuous variable that is not transformed.
`p`	The expected value of the numeric vector `bin.var` after dichotomization. Either `p` or `cutpoint` should be specified.
`cutpoint`	The value at which the vector `bin.var` should be dichotomized. Either `p` or `cutpoint` should be specified.

Value

The biserial correlation.

References

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

Demirtas, H. and Hedeker, D. (2016). Computing the point-biserial correlation under any underlying continuous distribution. *Communications in Statistics-Simulation and Computation*, **45(8)**, 2744-2751.

Examples

```
set.seed(123)
y1<-rweibull(n=100000, scale=1, shape=1.2)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=3, s2=1, pi=0.6)

pbs2bs(pbs=0.25, bin.var=y1, cont.var=y2, p=0.55)
pbs2bs(pbs=0.25, bin.var=y1, cont.var=y2, cutpoint=0.65484)
```

---

phi2tet *Computation of the Tetrachoric Correlation from the Phi Coefficient*

Description

This function computes the tetrachoric correlation between two continuous variables given the correlation after dichotomization of both variables (phi coefficient) as seen in Demirtas (2016). Before computation of the tetrachoric correlation, the specified phi coefficient is compared to the lower and upper correlation bounds for the two binary variables using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
phi2tet(phicoef, dist1, dist2)
```

Arguments

`phicoef`	The phi coefficient.
`dist1`	A list of length 3 containing the skewness, excess kurtosis, and expected value after dichotomization for the first continuous variable with names skewness, exkurtosis, and p, respectively.
`dist2`	A list of length 3 containing the skewness, excess kurtosis, and expected value after dichotomization for the second continuous variable with names skewness, exkurtosis, and p, respectively.

Value

The tetrachoric correlation.

References

Demirtas, H. (2016). A note on the relationship between the phi coefficient and the tetrachoric correlation under nonnormal underlying distributions. *The American Statistician*, **70(2)**, 143-148.

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

See Also

corrZ2corrY

Examples

```
set.seed(987)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=1)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=3, s2=1, pi=0.5)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

phi2tet(phicoef=0.1,
        dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=0.85),
        dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt, p=0.15))

phi2tet(phicoef=0.5,
        dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=0.10),
        dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt, p=0.30))
```

---

poly2ophi — *Computation of the Ordinal Phi Coefficient from the Polychoric Correlation*

Description

This function computes the ordinal phi coefficient between two variables after both of the variables are ordinalized given the correlation before ordinalization (polychoric correlation) as seen in Demirtas et al. (2016). Before computation of the ordinal phi coefficient, the specified polychoric correlation is compared to the lower and upper correlation bounds of the two continuous variables as defined by the respective skewness and excess kurtosis using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
poly2ophi(polycorr, dist1, dist2)
```

Arguments

`polycorr`	The polychoric correlation.
`dist1`	A list of length 3 containing the skewness, excess kurtosis, and a numeric vector of marginal probabilities for the first continuous variable with names skewness, exkurtosis, and p, respectively.
`dist2`	A list of length 3 containing the skewness, excess kurtosis, and a numeric vector of marginal probabilities for the second continuous variable with names skewness, exkurtosis, and p, respectively.

Value

The ordinal phi coefficient.

References

Demirtas, H., Ahmadian, R., Atis, S., Can, F.E., and Ercan, I. (2016). A nonnormal look at polychoric correlations: modeling the change in correlations before and after discretization. *Computational Statistics*, **31(4)**, 1385-1401.

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

Ferrari, P.A. and Barbiero, A. (2012). Simulating ordinal data. *Multivariate Behavioral Research*, **47(4)**, 566-589.

See Also

`corrY2corrZ`, `corrZ2ophi`, `mps2cps`

Examples

```
set.seed(567)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=3.6)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
```

```
y2<-gaussmix(n=100000, m1=0, s1=1, m2=2, s2=1, pi=0.3)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

poly2ophi(polycorr=0.5,
          dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=c(0.4, 0.3, 0.2, 0.1)),
          dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt , p=c(0.2, 0.2, 0.6)))

poly2ophi(polycorr=0.5,
          dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=c(0.1, 0.1, 0.1, 0.7)),
          dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt , p=c(0.8, 0.1, 0.1)))
```

---

pps2ps *Computation of the Polyserial Correlation from the Point-Polyserial Correlation*

Description

This function computes the polyserial correlation between two continuous variables given the correlation after ordinalization of one of the variables (point-polyserial correlation) as seen in Demirtas and Hedeker (2016). Before computation of the polyserial correlation, the specified point-polyserial correlation is compared to the lower and upper correlation bounds of the continuous variable and ordinalized variable using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
pps2ps(pps, ord.var, cont.var, cats, p=NULL, cutpoint=NULL)
```

Arguments

`pps`	The point-polyserial correlation.
`ord.var`	A numeric vector of the continuous variable before ordinalization.
`cont.var`	A numeric vector of the the continuous variable that is not transformed.
`cats`	A numeric vector of the categories in the ordinalization of `ord.var`.
`p`	A numeric vector of the marginal probabilities corresponding to each category in `cats`. The marginal probabilities must sum to 1. Either `p` or `cutpoint` should be specified.
`cutpoint`	A numeric vector of the cutpoints used to define the categories `cats`. Either `p` or `cutpoint` should be specified.

Value

The polyserial correlation.

References

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

Demirtas, H. and Hedeker, D. (2016). Computing the point-biserial correlation under any underlying continuous distribution. *Communications in Statistics-Simulation and Computation*, **45(8)**, 2744-2751.

See Also

ordY, mps2cps

Examples

```
set.seed(234)
y1<-rweibull(n=100000, scale=1, shape=25)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=2, s2=1, pi=0.5)

pps2ps(pps=0.3, ord.var=y1, cont.var=y2, cats=c(1,2,3,4), p=c(0.4, 0.3, 0.2, 0.1))
pps2ps(pps=0.3, ord.var=y1, cont.var=y2, cats=c(1,2,3,4), cutpoint=c(0.97341, 1.00750, 1.03421))
```

---

ps2pps *Computation of the Point-Polyserial Correlation from the Polyserial Correlation*

Description

This function computes the point-polyserial correlation between two variables after one of the variables is ordinalized given the correlation before ordinalization (polyserial correlation) as seen in Demirtas and Hedeker (2016). Before computation of the point-polyserial correlation, the specified polyserial correlation is compared to the lower and upper correlation bounds of the two continuous variables using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
ps2pps(ps, ord.var, cont.var, cats, p=NULL, cutpoint=NULL)
```

Arguments

`ps`	The polyserial correlation.
`ord.var`	A numeric vector of the continuous variable before ordinalization.
`cont.var`	A numeric vector of the the continuous variable that is not transformed.
`cats`	A numeric vector of the categories in the ordinalization of `ord.var`.

p	A numeric vector of the marginal probabilities corresponding to each category in cats. The marginal probabilities must sum to 1. Either p or cutpoint should be specified.
cutpoint	A numeric vector of the cutpoints used to define the categories in cats. Either p or cutpoint should be specified.

Value

The point-polyserial correlation.

References

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

Demirtas, H. and Hedeker, D. (2016). Computing the point-biserial correlation under any underlying continuous distribution. *Communications in Statistics-Simulation and Computation*, **45(8)**, 2744-2751.

See Also

ordY, mps2cps

Examples

```
set.seed(234)
y1<-rweibull(n=100000, scale=1, shape=25)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=2, s2=1, pi=0.5)

ps2pps(ps=0.6, ord.var=y1, cont.var=y2, cats=c(1,2,3,4), p=c(0.4, 0.3, 0.2, 0.1))
ps2pps(ps=0.6, ord.var=y1, cont.var=y2, cats=c(1,2,3,4), cutpoint=c(0.97341, 1.00750, 1.03421))
```

---

tet2phi *Computation of the Phi Coefficient from the Tetrachoric Correlation*

Description

This function computes the phi coefficient between two variables after both of the variables are dichotomized given the correlation before dichotomization (tetrachoric correlation) as seen in Demirtas (2016). Before computation of the phi coefficient, the specified tetrachoric correlation is compared to the lower and upper correlation bounds of the two continuous variables as defined by the respective skewness and excess kurtosis using the generate, sort and correlate (GSC) algorithm in Demirtas and Hedeker (2011).

Usage

```
tet2phi(tetcorr, dist1, dist2)
```

Arguments

`tetcorr`	The tetrachoric correlation.
`dist1`	A list of length 3 containing the skewness, excess kurtosis, and expected value after dichotomization for the first continuous variable with names skewness, exkurtosis, and p, respectively.
`dist2`	A list of length 3 containing the skewness, excess kurtosis, and expected value after dichotomization for the second continuous variable with names skewness, exkurtosis, and p, respectively.

Value

The phi coefficient.

References

Demirtas, H. (2016). A note on the relationship between the phi coefficient and the tetrachoric correlation under nonnormal underlying distributions. *The American Statistician*, **70(2)**, 143-148.

Demirtas, H. and Hedeker, D. (2011). A practical way for computing approximate lower and upper correlation bounds. *The American Statistician*, **65(2)**, 104-109.

See Also

`corrY2corrZ`, `corrZ2phi`

Examples

```
set.seed(987)
library(moments)

y1<-rweibull(n=100000, scale=1, shape=1)
y1.skew<-round(skewness(y1), 5)
y1.exkurt<-round(kurtosis(y1)-3, 5)

gaussmix <- function(n,m1,m2,s1,s2,pi) {
  I <- runif(n)<pi
  rnorm(n,mean=ifelse(I,m1,m2),sd=ifelse(I,s1,s2))
}
y2<-gaussmix(n=100000, m1=0, s1=1, m2=3, s2=1, pi=0.5)
y2.skew<-round(skewness(y2), 5)
y2.exkurt<-round(kurtosis(y2)-3, 5)

tet2phi(tetcorr=-0.4,
        dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=0.85),
        dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt, p=0.15))

tet2phi(tetcorr=0.7,
```

```
dist1=list(skewness=y1.skew, exkurtosis=y1.exkurt, p=0.10),
dist2=list(skewness=y2.skew, exkurtosis=y2.exkurt, p=0.30))
```

Index