Package ‘miscor’

April 2, 2017

Type Package
Title Miscellaneous Functions for the Correlation Coefficient
Version 0.1-1
Date 2017-04-02
Author Takuya Yanagida [cre, aut]
Maintainer Takuya Yanagida <takuya.yanagida@univie.ac.at>
Description Statistical test for the product-moment correlation coefficient based on H0: \( \rho = \rho_0 \)
including sample size computation, statistical test for comparing the product-moment
 correlation coefficient in independent and dependent samples, sequential triangular
test for the product-moment correlation coefficient, partial and semipartial correlation,
simulation of bivariate normal and non-normal distribution with a specified correlation.
License GPL-3
LazyLoad yes
LazyData true
Imports methods
Depends R (>= 3.0)
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2017-04-02 15:27:10 UTC

R topics documented:

comptest.cor ................................................................. 2
cnf.cor ................................................................. 4
par.cor ................................................................. 6
plot.seqtest .......................................................... 8
plot.sim.seqtest.cor .................................................. 9
print.comptest.cor ..................................................... 10
print.conf.cor ......................................................... 12
print.par.cor ......................................................... 13
Description

This function statistically compares product-moment correlation coefficients in independent and dependent samples.

Usage

comptest.cor(x = NULL, y = NULL, z = NULL, group = NULL,
            r.xy = NULL, r.xz = NULL, r.yz = NULL, n = NULL,
            r.1 = NULL, r.2 = NULL, n.1 = NULL, n.2 = NULL,
            alternative = c("two.sided", "less", "greater"),
            conf.level = 0.95, digits = 3, output = TRUE)

Arguments

- **x**
  - a numeric vector.
- **y**
  - a numeric vector.
- **z**
  - a numeric vector.
- **group**
  - a numeric vector indicating the group membership.
- **r.xy**
  - alternative specification, product-moment correlation coefficient between x and y.
- **r.xz**
  - alternative specification, product-moment correlation coefficient between x and z.
- **r.yz**
  - alternative specification, product-moment correlation coefficient between y and z.
- **n**
  - alternative specification, number of observations.
- **r.1**
  - alternative specification, product-moment correlation coefficient in group 1.
- **r.2**
  - alternative specification, product-moment correlation coefficient in group 2.
n.1 alternative specification, number of observations in group 1.
n.2 alternative specification, number of observations in group 2.
alternative a character string describing the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". 
conf.level confidence level of the interval.
digits integer indicating the number of decimal places to be displayed.
output logical: if TRUE, output is shown.

Details
In dependent samples, the function tests the two-sided null hypothesis H0: \( \rho_{xy} = \rho_{xz} \) or the one-sided null hypothesis H0: \( \rho_{xy} \geq \rho_{xz} \) or \( \rho_{xy} \leq \rho_{xz} \). Function parameters are specified using either \((x, y, z)\) or \((r_{xy}, r_{xz}, r_{yz}, n)\). In independent samples, the function tests the two-sided null hypothesis H0: \( \rho_1 = \rho_2 \) or the one-sided null hypothesis H0: \( \rho_1 \geq \rho_2 \) or \( \rho_1 \leq \rho_2 \). Function parameters are specified using either \((x, y, group)\) or \((r_1, r_2, n_1, n_2)\).

Value
Returns an object of class comptest.cor with following entries:
call function call
dat data.frame with x, y and z (if available)
spec specification of function arguments
res list with results depending on the analysis (independent of dependent samples), i.e., \( z \) (test statistic), pval (significance value), \( r_{xy}, r_{xz}, r_{yz}, r_1, r_2, \) diff (difference), n, n_1, n_2, lower (lower limit of CI), upper (upper limit of CI).

Author(s)
Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References

See Also
test.cor, seqtest.cor

Examples
# Dependent samples: Generate random data
x <- c(3, 2, 2, 3, 7, 8, 5, 9)
y <- c(2, 4, 1, 5, 7, 3, 6, 7)
z <- c(1, 4, 3, 3, 1, 4, 2, 5)
conf.cor

#--------------------------------------
# Dependent samples
# H0: rho.xy == rho.xz,  H1: rho.xy != rho.xz
comptest.cor(x, y, z)

#--------------------------------------
# Dependent samples
# H0: rho.xy <= rho.xz,  H1: rho.xy > rho.xz
# r.xy = 0.44, r.xz = 0.21, r.yz = 0.20, n = 120
comptest.cor(r.xy = 0.44, r.xz = 0.21, r.yz = 0.20, n = 120,
             alternative = "greater")

#--------------------------------------
# Independent samples: Generate random data
dat <- data.frame(group = rep(1:2, each = 200),
                   rbind(sim.cor(200, rho = 0.3),
                         sim.cor(200, rho = 0.5)))

#--------------------------------------
# Independent samples
# H0: rho.1 == rho.2,  H1: rho.1 != rho.2
comptest.cor(x = dat$x, y = dat$y, group = dat$group)

#--------------------------------------
# Independent samples
# H0: rho.1 >= rho.2,  H1: rho.1 !< rho.2
# Group 1:  r = 0.32, n = 108
# Group 2:  r = 0.56, n = 113
comptest.cor(r.1 = 0.32, n.1 = 108, r.2 = 0.56, n.2 = 113,
             alternative = "less")

---

**Description**

This function computes the product-moment correlation coefficient with two-sided or one-sided confidence interval using Fisher's z tranformation.

**Usage**

```r
conf.cor(x = NULL, y = NULL, r = NULL, n = NULL,
         alternative = c("two.sided", "less", "greater"),
         conf.level = 0.95, digits = 3, output = TRUE)
```
Arguments

x a numeric vector.

y a numeric vector.

r alternative specification, product-moment correlation coefficient.

n alternative specification, number of observations.

alternative a character string describing the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".

conf.level confidence level of the interval.

digits integer indicating the number of decimal places to be displayed.

output logical; if TRUE, output is shown.

Value

Returns an object of class conf.cor with following entries:

call function call

dat data.frame with x and y (if available)

spec specification of function arguments

res list with results, i.e., r (correlation coefficient), n, lower (lower limit of CI), upper (upper limit of CI)

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References


See Also

test.cor, seqtest.cor, comptest.cor

Examples

#--------------------------------------
# Two-sided 95% Confidence Interval
# r = 0.55, n = 120
#--------------------------------------

conf.cor(r = 0.55, n = 120)

#--------------------------------------
# One-sided 99% Confidence Interval
par.cor

# Generate random data
dat <- sim.cor(100, rho = 0.4)

conf.cor(dat$x, dat$y, conf.level = 0.99, alternative = "less")

### par.cor

**Partial and semipartial correlation**

**Description**

This function computes the partial or semipartial correlation coefficient between two variables. In addition, this function can test the partial or semipartial correlation coefficient for H0: \( \rho_p = \rho_0 \), so that any value for \( \rho_0 \) can be specified.

**Usage**

```r
dat <- sim.cor(100, rho = 0.4)

corr <- cor(dat$x, dat$y)

par.cor(x = NULL, y = NULL, p.xy = NULL, p.x = NULL, p.y = NULL,
sig = FALSE, rho0 = 0, alternative = c("two.sided", "less", "greater"),
reduced = FALSE, conf.level = 0.95, digits = 3, output = TRUE)
```

**Arguments**

- **x**: a numeric vector.
- **y**: a numeric vector.
- **p.xy**: a numeric vector or data.frame, variable(s) residualized from x and y.
- **p.x**: a numeric vector or data.frame, variable(s) residualized only from x.
- **p.y**: a numeric vector or data.frame, variable(s) residualized only from y.
- **sig**: logical: if TRUE, statistical significance test is conducted.
- **rho0**: a number indicating \( \rho_0 \), the value under the null hypothesis.
- **alternative**: a character string describing the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
- **reduced**: logical: if TRUE, computation is based on the reduced formula.
- **conf.level**: confidence level of the interval.
- **digits**: integer indicating the number of decimal places to be displayed.
- **output**: logical: if TRUE, output is shown.

**Details**

Partial correlation is the correlation of \( x \) and \( y \) while statistically controlling for third variables specified in the argument \( p.xy \). These variables are residualized from \( x \) and \( y \) using (multiple) regression models. Semipartial correlation is the correlation of \( x \) and \( y \) while statistically controlling for third variables only for \( x \) (specified in the argument \( p.x \)) or \( y \) (specified in the argument \( p.y \)). These variables are residualized from \( x \) or \( y \) using a (multiple) regression model.

**Value**

Returns an object of class \texttt{par.cor} with following entries:
Function call:

- `par.cor` function
- Argument `dat` specifies the list with data for `x.resid` (x residualized), `y.resid` (y residualized), x, y, p.xy, p.y, and p.x
- Argument `spec` specifies the function argument method
- Argument `res` specifies the list with results, i.e., t or z (test statistic), df (degree of freedom) pval (significance value), r.p (partial or semipartial correlation coefficient), n (sample size), lower (lower limit of CI), upper (upper limit of CI)

Author(s):

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References:


See Also:

test.cor, conf.cor, comptest.cor, seqtest.cor

Examples:

```r
dat <- data.frame(x = c(4, 6, 8, 8, 9, 4),
                  y = c(3, 7, 9, 9, 8, 9),
                  z = c(1, 3, 4, 4, 5, 2))

# Partial correlation
par.cor(dat$x, dat$y, p.xy = dat$z)

# Semipartial correlation
# remove z from x
par.cor(dat$x, dat$y, p.x = dat$z)

# Semipartial correlation
# remove z from y
par.cor(dat$x, dat$y, p.y = dat$y)

# Partial correlation: Two-sided test
# H0: rho.p == 0, H1: rho.p != 0
par.cor(dat$x, dat$y, p.xy = dat$z, sig = TRUE)

# Partial correlation: One-sided test
# H0: rho.p <= 0.2, H1: rho.p > 0.2
```
par.cor(dat$x, dat$y, p.xy = dat$z, sig = TRUE,
       rho0 = 0.4, alternative = "less")

plot.seqtest

Description
This function plots the seqtest object

Usage
## S3 method for class 'seqtest'
plot(x, ...)

Arguments
  x    seqtest object
  ...  further arguments passed to or from other methods

Author(s)
Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

See Also
seqtest.cor, print.seqtest

Examples
#--------------------------------------
# Sequential triangular test for the product-moment correlation coefficient

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2,
                        alpha = 0.05, beta = 0.2)

plot(seq.obj)
Description

This function plots the sim.seqtest.cor object

Usage

## S3 method for class 'sim.seqtest.cor'
plot(x, plot.lines = TRUE, plot.nom = TRUE,
     ylim = NULL, type = "b", pch = 19, lty = 1, lwd = 1, ...)

Arguments

x sim.seqtest.cor object.
plot.lines plot lines connecting points with the x- and y-axis.
plot.nom plot line at the nominal alpha.
ylim the y limits of the plot.
type what type of plot should be drawn ("p" for points, "l" for lines and "b" for both).
pch plotting character.
lty line type.
lwd line width.
... further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References


See Also

sim.seqtest.cor, seqtest.cor
print.comptest.cor

Examples

## Not run:

```
#---------------------------------------------
# Determine optimal k and nominal type-II-risk
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.01, beta = 0.05, delta = 0.25

# Step 1: Determine the optimal size of subsamples (k)

sim.obj.1 <- sim.seqtest.cor(rho.sim = 0.3, k = seq(4, 16, by = 1), rho = 0.3,
                          alternative = "greater",
                          delta = 0.25, alpha = 0.05, beta = 0.05,
                          runs = 10000)

plot(sim.obj.1)

# Step 2: Determine the optimal nominal type-II-risk based on
# the optimal size of subsamples (k) from step 1

sim.obj.2 <- sim.seqtest.cor(rho.sim = 0.55, k = 16, rho = 0.3,
                          alternative = "greater",
                          delta = 0.25, alpha = 0.05, beta = seq(0.05, 0.15, by = 0.01),
                          runs = 10000)

plot(sim.obj.2)

## End(Not run)
```

---

print.comptest.cor  

**Print comptest.cor**

Description

This function prints the comptest.cor object

Usage

```r
## S3 method for class 'comptest.cor'
print(x, ...)
```

Arguments

- **x**: comptest.cor object.
- **...**: further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>
References


See Also

comptest.cor

Examples

```r
# Dependent samples
# Generate random data
x <- c(3, 2, 2, 3, 7, 8, 5, 9)
y <- c(2, 4, 1, 5, 7, 3, 6, 7)
z <- c(1, 4, 3, 3, 1, 4, 2, 5)

# H0: rho.xy = rho.xz, H1: rho.xy != rho.xz
obj <- comptest.cor(x, y, z, output = FALSE)
print(obj)

# H0: rho.xy <= rho.xz, H1: rho.xy > rho.xz
# r.xy = 0.44, r.xz = 0.21, r.yz = 0.20, n = 120
obj <- comptest.cor(r.xy = 0.44, r.xz = 0.21, r.yz = 0.20, n = 120,
alternative = "greater", output = FALSE)
print(obj)

# Independent samples
# Generate random data
dat <- data.frame(group = rep(1:2, each = 200),
                 rbind(sim.cor(200, rho = 0.3),
                       sim.cor(200, rho = 0.5)))

# H0: rho.1 = rho.2, H1: rho.1 != rho.2
obj <- comptest.cor(x = dat$x, y = dat$y, group = dat$group,
                   output = FALSE)
print(obj)

# H0: rho.1 >= rho.2, H1: rho.1 < rho.2
```

# Group 1: \( r = 0.32, n = 108 \)
# Group 2: \( r = 0.56, n = 113 \)

```r
obj <- comptest.cor(r.1 = 0.32, n.1 = 108, r.2 = 0.56, n.2 = 113,
                    alternative = "less", output = FALSE)
print(obj)
```

---

### Description

This function prints the `cor.conf` object.

### Usage

```r
## S3 method for class 'cor.conf'
print(x, ...)
```

### Arguments

- `x` : cor.conf object.
- `...` : further arguments passed to or from other methods.

### Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

### References


Kubinger, K. D., Rasch, D., & Simeckova, M. (2007). Testing a correlation coefficient's significance: Using \( H_0: 0 < \rho \leq \lambda \) is preferable to \( H_0: \rho = 0 \). *Psychology Science, 49,* 74-87.

### See Also

`conf.cor`

### Examples

```r
#--------------------------------------
# Two-sided 95% Confidence Interval
# \( r = 0.55, n = 120 \)

obj <- conf.cor(r = 0.55, n = 120, output = FALSE)
print(obj)
```
#-----------------------------
# One-sided 99% Confidence Interval
#
# Generate random data
dat <- sim.cor(100, rho = 0.4)
#
# Generate random data
obj <- conf.cor(dat$x, dat$y, conf.level = 0.99, alternative = "less",
output = FALSE)
print(obj)

print.par.cor  

Description
This function prints the par.cor object

Usage
### S3 method for class 'par.cor'
print(x, ...)

Arguments
x  par.cor object.
...

further arguments passed to or from other methods.

Author(s)
Takuya Yanagida <takuya.yanagida@univie.ac.at>

References
New York: John Wiley & Sons.

See Also
par.cor

Examples
dat <- data.frame(x = c(4, 6, 8, 8, 9, 4),
    y = c(3, 7, 9, 8, 9, 3),
    z = c(1, 3, 4, 4, 5, 2))

#-----------------------------
# Partial correlation
# S3 method for class 'seqtest'
print(x, ...)
print.sim.seqtest.cor

Author(s)
Takuya Yanagida <takuya.yanagida@univie.ac.at>

References

See Also
seqtest.cor, plot.seqtest

Examples
#--------------------------------------
# Sequential triangular test for product-moment correlation coefficient
seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2,
alpha = 0.05, beta = 0.2, output = FALSE)
print(seq.obj)

Description
This function prints the sim.seqtest.cor object

Usage
## S3 method for class 'sim.seqtest.cor'
print(x, ...)

Arguments
x sim.seqtest.cor object.
... further arguments passed to or from other methods.

Author(s)
Takuya Yanagida <takuya.yanagida@univie.ac.at>
References


See Also

`sim.seqtest.cor, plot.sim.seqtest.cor`

Examples

```r
## Not run:

#----------------------------------------------------------
# Determine optimal k and nominal type-II-risk
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.01, beta = 0.05, delta = 0.25

# Step 1: Determine the optimal size of subsamples (k)

sim.obj <- sim.seqtest.cor(rho.sim = 0.3, k = seq(4, 16, by = 1), rho = 0.3,
                           alternative = "greater",
                           delta = 0.25, alpha = 0.05, beta = 0.05,
                           runs = 10000, output = FALSE)

print(sim.obj)

# Step 2: Determine the optimal nominal type-II-risk based on
# the optimal size of subsamples (k) from step 1

sim.obj <- sim.seqtest.cor(rho.sim = 0.55, k = 16, rho = 0.3,
                           alternative = "greater",
                           delta = 0.25, alpha = 0.05, beta = seq(0.05, 0.15, by = 0.01),
                           runs = 10000, output = FALSE)

print(sim.obj)

## End(Not run)
```

print.size

`print.size` is a function that prints the size object.

Description

This function prints the size object.

Usage

```r
## S3 method for class 'size'
print(x, ...)
```
Arguments

x size object.

... further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References


See Also

size.cor

Examples

```r
#--------------------------------------
# H0: rho = 0.3, H1: rho != 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2

n <- size.cor(delta = 0.2, rho = 0.3, alpha = 0.05, beta = 0.2, output = FALSE)

print(n)
```

Description

This function prints the test.cor object

Usage

```r
## S3 method for class 'test.cor'
print(x, ...)
```

Arguments

x test.cor object.

... further arguments passed to or from other methods.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>
References


See Also
test.cor

Examples

```r
# Two-sided test
# H0: rho == 0, H1: rho != 0
# r = 0.23, n = 60

obj <- test.cor(r = 0.23, n = 120, output = FALSE)
print(obj)

# Two-sided test
# H0: rho == 0.4, H1: rho != 0.4
# r = 0.55, n = 120

obj <- test.cor(r = 0.55, n = 120, rho0 = 0.4,
                 output = FALSE)
print(obj)

# One-sided test
# H0: rho <= 0.4, H1: rho > 0.4

# Generate random data
dat <- sim.cor(100, rho = 0.4)

obj <- test.cor(dat$x, dat$y, rho0 = 0.4, output = FALSE)
print(obj)
```

scatterplot  *Scatterplot Matrices*

Description

This function produces a scatterplot matrix for integer data

Usage

```r
scatterplot(dat, type = c("jitter", "size", "count", "sun", "identity"),
            barplot = TRUE, curves = TRUE)
```
Arguments

- **dat**: a data frame
- **type**: type of plot, i.e., 'jitter', 'size', 'count', 'sun', and 'identity'
- **barplot**: logical: if TRUE barplots are shown in the diagonals.
- **curves**: logical: if TRUE lowess smoothing curves are added in the upper diagonal.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>

References


See Also

test.cor, seqtest.cor

Examples

dat <- round(sim.cor(200, rho = 0.7))
# Scatterplot matrix: jitter
scatterplot(dat)

# Scatterplot matrix: size
scatterplot(dat, type = "size")

# Scatterplot matrix: count
scatterplot(dat, type = "count")

# Scatterplot matrix: sun
scatterplot(dat, type = "sun")

dat <- roundHsimNcorHRPPL rho = PN7II
C scatterplot matrixZ jitter
scatterplotHdatI
C scatterplot matrixZ size
scatterplotHdatL type = BsizeBI
C scatterplot matrixZ count
scatterplotHdatL type = BcountBI
C scatterplot matrixZ sun
scatterplotHdatL type = BsunBI

seqtest.cor

Sequential triangular test for the product-moment correlation coefficient

Description

This function performs the sequential triangular test for the product-moment correlation coefficient.

Usage

seqtest.cor(x, k, rho,
alternative = c("two.sided", "less", "greater"),
delta, alpha = 0.05, beta = 0.1,
output = TRUE, plot = FALSE)
Arguments

- **x**: initial data, i.e., product-moment correlation coefficient in a sub-sample of k observations.
- **k**: number of observations in each sub-sample.
- **rho**: a number indicating the correlation under the null hypothesis, $\rho_0$.
- **alternative**: a character string specifying the alternative hypothesis.
- **delta**: minimum difference to be detected, $\delta$.
- **alpha**: type-I-risk, $\alpha$.
- **beta**: type-II-risk, $\beta$.
- **output**: logical: if TRUE, output is shown.
- **plot**: logical: if TRUE, an initial plot is generated.

Details

Null and alternative hypothesis is specified using arguments **rho** and **delta**. Note that the argument **k** (i.e., number of observations in each sub-sample) has to be specified. At least k = 4 is needed. The optimal value of k should be determined based on statistical simulation using **sim.seqtest.cor** function.

In order to specify a one-sided test, argument **alternative** has to be used (i.e., two-sided tests are conducted by default). That is, **alternative = "less"** specifies the null hypothesis, $H_0$: $\rho \geq \rho_0$ and the alternative hypothesis, $H_1$: $\rho < \rho_0$; **alternative = "greater"** specifies the null hypothesis, $H_0$: $\rho \leq \rho_0$ and the alternative hypothesis, $H_1$: $\rho > \rho_0$.

The main characteristic of the sequential triangular test is that there is no fixed sample size given in advance. That is, for the most recent sampling point, one has to decide whether sampling has to be continued or either the null- or the alternative hypothesis can be accepted given specified precision requirements (i.e. type-I-risk, type-II-risk and an effect size). The sequence of data pairs must we split into sub-samples of length k >= 4 each. The (cumulative) test statistic $Z_m$ on a Cartesian coordinate system produces a "sequential path" on a continuation area as a triangle. As long as the statistic remains within that triangle, additional data have to be sampled. If the path touches or exceeds the borderlines of the triangle, sampling is completed. Depending on the particular borderline, the null-hypothesis is either accepted or rejected.

Value

Returns an object of class **seqtest**, to be used for later update steps. The object has following entries:

- **call**: function call
- **type**: type of the test (i.e., correlation coefficient)
- **spec**: specification of function arguments
- **tri**: specification of triangular
- **dat**: data
- **res**: list with results
**sim.cor**

Simulate bivariate distribution with a specified correlation

**Description**

This function simulates bivariate distribution with correlation equal to $\rho$, mean equal to $\text{mean}$, standard deviation equal to $\text{sd}$, skewness equal to $\text{skewness}$, and kurtosis equal to $\text{kurtosis}$ by Fleishman polynomials. Note that the specified skewness and kurtosis parameters have to be in line with $\text{kurtosis} \geq \left(\text{skewness}^2 - 2\right)$

**Usage**

\[
\text{sim.cor}(n, \rho, \text{mean} = c(0, 0), \text{sd} = c(1, 1), \\
\quad \text{skewness} = c(0, 0), \text{kurtosis} = c(0, 0))
\]
Arguments

n       number of observations.
rho     correlation.
mean    mean vector.
sd      standard deviation vector.
skewness skewness vector.
kurtosis kurtosis vector.

Value

Returns a data.frame with variables x and y.

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References


See Also

test.cor, seqtest.cor, comptest.cor.

Examples

```
#-----------------------------------------------
# Bivariate distribution with rho = 0.3, n = 200
# x: skewness = 0, kurtosis = 0
# y: skewness = 0, kurtosis = 0

sim.cor(200, rho = 0.3)
```

```
#-----------------------------------------------
# Bivariate distribution with rho = 0.4, n = 500
# x: skewness = 0, kurtosis = 1.5
# y: skewness = 2, kurtosis = 7

sim.cor(500, rho = 0.4, skewness = c(0, 1.5), kurtosis = c(2, 7))
```
Description

This function performs a statistical simulation for the sequential triangular test for the product-moment correlation coefficient.

Usage

simNseqtestNcorHrhoNsimL kL rhoL
alternative = c("two.sided", "less", "greater"),
delta, alpha = 0.05, beta = 0.1, runs = 1000,
m.x = 0, sd.x = 1, m.y = 0, sd.y = 1,
digits = 3, output = TRUE, plot = FALSE)

Arguments

rho.sim simulated population correlation coefficient, \( \rho \).

k an integer or a numerical vector indicating the number of observations in each sub-sample.

rho a number indicating the correlation under the null hypothesis, \( \rho_0 \).

alternative a character string specifying the alternative hypothesis.

delta minimum difference to be detected, \( \delta \).

alpha type-I-risk, \( \alpha \).

beta an integer or a numerical vector indicating the type-II-risk, \( \beta \).

runs number of simulation runs.

m.x population mean of simulated vector x.

sd.x population standard deviation of simulated vector x.

m.y population mean of simulated vector y.

sd.y population standard deviation of simulated vector y.

digits integer indicating the number of decimal places to be displayed.

output logical: if TRUE, output is shown.

plot logical: if TRUE, plot is shown.

Details

In order to determine the optimal k, simulation is conducted under the H0 condition, i.e., \( \rho_{\text{sim}} = \rho \). Simulation is carried out for a sequence of k values to seek for the optimal k where the empirical alpha is as close as possible to the nominal alpha. In order to determine optimal beta (with fixed k), simulation is conducted under the H1 condition, i.e., \( \rho_{\text{sim}} = \rho + \delta \) or
rho.sim = rho - delta. Simulation is carried out for a sequence of beta values to seek for the optimal beta where the empirical beta is as close as possible to the nominal beta.

In order to specify a one-sided test, argument alternative has to be used (i.e., two-sided tests are conducted by default). Specifying argument alternative = "less" conducts the simulation for the null hypothesis, H0: \( \rho \geq \rho_0 \) with the alternative hypothesis, H1: \( \rho < \rho_0 \); specifying argument alternative = "greater" conducts the simulation for the null hypothesis, H0: \( \rho \leq \rho_0 \) with the alternative hypothesis, H1: \( \rho > \rho_0 \).

Value

Returns an object of class sim.seqtest.cor with following entries:

- **call**: function call
- **spec**: specification of function arguments
- **simres**: list with results (for each k or beta) for each run
- **res**: data.frame with results, i.e., k, alpha.nom (nominal alpha), alpha.emp (estimated empirical alpha), beta.nom (nominal beta), delta, decision = H0, p.H1 (proportion decision = H1), A VN (average number of V), ASN (average number of sample pairs)

Author(s)

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References


See Also

seqtest.cor, plot.sim.seqtest.cor, print.sim.seqtest.cor

Examples

```r
## Not run:
#-----------------------------------------------------
# Determine optimal k and nominal type-II-risk
# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.01, beta = 0.05, delta = 0.25

# Step 1: Determine the optimal size of subsamples (k)

sim.seqtest.cor(rho.sim = 0.3, k = seq(4, 16, by = 1), rho = 0.3, alternative = "greater", delta = 0.25, alpha = 0.05, beta = 0.05, runs = 10000, plot = TRUE)

# Step 2: Determine the optimal nominal type-II-risk based on
# the optimal size of subsamples (k) from step 1

sim.seqtest.cor(rho.sim = 0.55, k = 16, rho = 0.3, alternative = "greater",
```


```r
delta = 0.25, alpha = 0.05, beta = seq(0.05, 0.15, by = 0.01),
runs = 10000, plot = TRUE)

## End(Not run)
```

---

**size.cor**

Sample size determination for testing the product-moment correlation coefficient

---

**Description**

This function performs sample size computation for testing the product-moment correlation coefficient for \( H_0: \rho = \rho_0 \) based on precision requirements (i.e., type-I-risk, type-II-risk and an effect size).

**Usage**

```r
size.cor(rho = NULL, delta, alternative = c("two.sided", "less", "greater"),
alpha = 0.05, beta = 0.1, output = TRUE)
```

**Arguments**

- **rho**: a number indicating the correlation coefficient under the null hypothesis, \( \rho_0 \).
- **delta**: minimum difference to be detected, \( \delta \).
- **alternative**: a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
- **alpha**: type-I-risk, \( \alpha \).
- **beta**: type-II-risk, \( \beta \).
- **output**: logical: if TRUE, output is shown.

**Value**

Returns an object of class size with following entries:

- **call**: function call
- **type**: type of the test (i.e., correlation coefficient)
- **spec**: specification of function arguments
- **res**: list with the result, i.e., optimal sample size

**Author(s)**

Takuya Yanagida <takuya.yanagida@univie.ac.at>,

---

```r
```

---

```r
```
References


See Also
testNcor, seqtestNcor

Examples

```
# H0: rho = 0.3, H1: rho != 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2
sizeNcor(rho = 0.3, delta = 0.2, alpha = 0.05, beta = 0.2)

# H0: rho <= 0.3, H1: rho > 0.3
# alpha = 0.05, beta = 0.2, delta = 0.2
sizeNcor(rho = 0.3, delta = 0.2, alternative = "greater",
         alpha = 0.05, beta = 0.2)
```

testNcor

Test for the product-moment correlation coefficient for H0: \( \rho = \rho_0 \)

Description

This function tests the product-moment correlation coefficient for H0: \( \rho = \rho_0 \), so that any value for \( \rho_0 \) can be specified.

Usage

```
test.cor(x = NULL, y = NULL, r = NULL, n = NULL, rho0 = 0,
         alternative = c("two.sided", "less", "greater"), reduced = FALSE,
         conf.level = 0.95, digits = 3, output = TRUE)
```

Arguments

- **x**: a numeric vector.
- **y**: a numeric vector.
- **r**: alternative specification, product-moment correlation coefficient.
- **n**: alternative specification, number of observations.
- **rho0**: a number indicating \( \rho_0 \), the value under the null hypothesis.
alternative a character string describing the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".

reduced logical: if TRUE, computation is based on the reduced formula.

conf.level confidence level of the interval.

digits integer indicating the number of decimal places to be displayed.

output logical: if TRUE, output is shown.

Details
Computation is based on Fisher's z transformation $z = 0.5 \cdot \ln \left( \frac{1+r}{1-r} \right)$. The difference between the full formula (i.e., reduced = FALSE) and the reduced formula (i.e., reduced = TRUE) is that the full formula includes the term $\frac{\rho}{n-1}$ in the formula of the expectation $E$, i.e.,

$$E(z) = 0.5 \cdot \ln \left( \frac{1+\rho}{1-\rho} \right) + \frac{\rho}{n-1}$$

whereas the reduced formula does not include this term, i.e.,

$$E(z) = 0.5 \cdot \ln \left( \frac{1+\rho}{1-\rho} \right)$$

It is recommended to always use the full formula, especially in small samples.

Value
Returns an object of class test.cor with following entries:

call function call

dat data.frame with x and y (if available)

spec specification of function arguments

res list with results, i.e., t or z (test statistic), df (degree of freedom), pval (significance value), r (correlation coefficient), n (sample size), lower (lower limit of CI), upper (upper limit of CI)

Author(s)
Takuya Yanagida <takuya.yanagida@univie.ac.at>,

References


Kubinger, K. D., Rasch, D., & Simeckova, M. (2007). Testing a correlation coefficient's significance: Using H0: $0 < \rho \leq \lambda$ is preferable to H0: $\rho = 0$. *Psychology Science*, 49, 74-87.

SeeAlso
size.cor, compost.test.cor, seqtest.cor
Examples

#--------------------------------------
# Two-sided test
# H0: rho == 0, H1: rho != 0
# r = 0.23, n = 60

test.cor(r = 0.23, n = 120)

#--------------------------------------
# Two-sided test
# H0: rho == 0.4, H1: rho != 0.4
# r = 0.55, n = 120

test.cor(r = 0.55, n = 120, rho0 = 0.4)

#--------------------------------------
# One-sided test
# H0: rho <= 0.4, H1: rho > 0.4

# Generate random data
dat <- sim.cor(100, rho = 0.4)
test.cor(dat$x, dat$y, rho0 = 0.4)

update.seqtest

update.seqtest

update.seqtest

Description

This function updates the seqtest object

Usage

## S3 method for class 'seqtest'
update(object, x = NULL, y = NULL, initial = FALSE,
       output = TRUE, plot = TRUE, ...)

Arguments

- object: cor.seqtest object.
- x: data for group 1.
- y: data for group 2.
- initial: logical, used internally for creating a seqtest object
- output: logical: if TRUE, output is shown.
- plot: logical: if TRUE, plot is shown.
- ...: further arguments passed to or from other methods.
Author(s)
Takuya Yanagida <takuya.yanagida@univie.ac.at>

References
Schneider, B., Rasch, D., Kubinger, K. D., & Yanagida, T. (2015). A Sequential triangular test of a correlation coefficient’s null-hypothesis: 0 < \( \rho \) \( \leq \) \( \rho_0 \). Statistical Papers, 56, 689-699.

See Also
seqtest.cor,

Examples

```r
#-----------------------------
# Sequential triangular test for the product-moment correlation coefficient

seq.obj <- seqtest.cor(0.46, k = 14, rho = 0.3, delta = 0.2, alpha = 0.05, beta = 0.2, plot = TRUE)

seq.obj <- update(seq.obj, c(0.56, 0.76, 0.56, 0.52))
```

```r
#-----------------------------
```
Index

comptest.cor, 2, 5, 7, 11, 22, 27
conf.cor, 4, 7, 12

par.cor, 6, 13
plot.seqtest, 8, 15
plot.sim.seqtest.cor, 9, 16, 24
print.comptest.cor, 10
print.conf.cor, 12
print.par.cor, 13
print.seqtest, 8, 14
print.sim.seqtest.cor, 15, 24
print.size, 16
print.test.cor, 17

scatterplot, 18
seqtest.cor, 3, 5, 7-9, 15, 19, 19, 22, 24, 26, 27, 29
sim.cor, 21
sim.seqtest.cor, 9, 16, 20, 21, 23
size.cor, 17, 25, 27

test.cor, 3, 5, 7, 18, 19, 22, 26, 26

update.seqtest, 21, 28