Package: zeroEQpart (via r-universe)

September 7, 2024

Type Package

Title Zero Order vs (Semi) Partial Correlation Test and CI

Version 0.1.0

Imports ppcor, stats, utils, MASS

Description Uses bootstrap to test zero order correlation being equal to a partial or semi-partial correlation (one or two tailed). Confidence intervals for the parameter (zero order minus partial) can also be determined. Implements the bias-corrected and accelerated bootstrap method as described in ``An Introduction to the Bootstrap'' Efron (1983) <0-412-04231-2>.

URL https://github.com/djrichar92/zeroEQpart

BugReports https://github.com/djrichar92/zeroEQpart/issues

License GPL-2

Encoding UTF-8

LazyData FALSE

Date 2018-05-17

RoxygenNote 6.1.0

Repository https://djrichar92.r-universe.dev

RemoteUrl https://github.com/djrichar92/zeroeqpart

RemoteRef HEAD

RemoteSha 7d4ad5902404b6363198737dc2b5c339ba5f8ecc

Contents

zeroEQpart-package *Zero Order vs (Semi) Partial Correlation Test and CI*

Description

Calculate the statistical significance of a zero order correlation being equal to a partial or semi-partial correlation using the bias-corrected and accelerated bootstrap method from "An Introduction to the Bootstrap" Efron (1983) <0-412-04231-2>. Confidence intervals for the parameter (zero order minus partial) can also be determined.

pzcor

The `pzcor` function tests one of the following null hypotheses:

- $\rho.xy - \rho.xy.z = 0$ (default)
- $\rho.xy - \rho.xy.z \geq 0$
- $\rho.xy - \rho.xy.z \leq 0$

See `pzcor` for details.

pzconf

The `pzconf` function computes confidence intervals for the parameter: $\rho.xy - \rho.xy.z$. To be used with `pzcor`. See `pzconf` for details.

Author(s)

Maintainer: Dan Richard <`djrichar@ualberta.ca`>

Other contributors:

- Karen Buro <`BuroK@macewan.ca`> [contributor]
- Wanhua Su <`SuW3@macewan.ca`> [contributor]

See Also

Useful links:

- `https://github.com/djrichar92/zeroEQpart`
- Report bugs at `https://github.com/djrichar92/zeroEQpart/issues`

pzconf — *Calculate Confidence Intervals for the Difference of Zero Order and (Semi) Partial Correlation*

Description

The pzconf function calculates confidence intervals for a zero order correlation minus a (semi) partial correlation ($\rho.xy - \rho.xy.z$). It is intended to be used after the pzcor function.

Usage

```
pzconf(pzcor_obj, level = 0.9)
```

Arguments

pzcor_obj	pzcor object (output from pzcor function).
level	numerical. Confidence level used to calculate the confidence interval. This may be a vector so multiple intervals can be determined.

Details

The pzconf function calculates confidence intervals based on the bootstrap distribution determined from the pzcor function. See ?pzcor for details.

Value

The confidence interval(s) is(are) displayed in a dataframe with four columns: Level, Lower, Upper, and Warnings. Level refers to the confidence level of the interval. Lower and Upper are the respective lower and upper bounds of the interval. Warnings may say "Max Level Passed" to show that the specified confidence level exceeds the largest confidence interval that can be determined from the test. The largest confidence interval is shown in the last row (named "Max").

See Also

pzcor

Examples

```
require(graphics)
require(MASS)
# data
set.seed(1111)
mu <- rep(0,4)
Sigma <- matrix(.2, nrow=4, ncol=4) + diag(4)*.8
data <- mvrnorm(n=100, mu=mu, Sigma=Sigma)

# p.(1,2) = p.(1,2)|(3,4) test
test <- pzcor(data[,1], data[,2], data[,c(3,4)], k = 2000)
```

```
hist(test$distribution)
pzconf(test, c(0.9, 0.95, 0.99))
```

pzcor *Test for Equal Zero Order and (Semi) Partial Correlation*

Description

Compute a bootstrap test to determine whether zero order correlation is equal to partial or semi-partial correlation.

Usage

```
pzcor(x, y, z, semi = FALSE, k = 1000, method = "pearson",
  test = "eq")
```

Arguments

x	a numeric vector.
y	a numeric vector.
z	a numeric vector (data.frame, matrix, etc.)
semi	logical. If TRUE, then the semi-partial correlation between x and y given z is used. If FALSE (default), then the partial correlation between x given z and y given z is used.
k	the number of bootstrap samples taken (default is 1000).
method	a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman" can be abbreviated.
test	character string denoting the null hypothesis to be tested. Can be one of the three: • 'eq' tests $\rho.xy - \rho.xy.z = 0$ (default) • 'gt' tests $\rho.xy - \rho.xy.z \geq 0$ • 'lt' tests $\rho.xy - \rho.xy.z \leq 0$

Details

Uses the bias-corrected and accelerated (BCa) bootstrap method to test if the difference $\rho.xy - \rho.xy.z$ is equal to, above, or below zero where $\rho.xy$ is the zero order correlation between variables x and y, and $\rho.xy.z$ is the (semi) partial correlation between the respective variables after partialing out variables represented by z.

If the bootstrap distribution of $\rho.xy - \rho.xy.z$ is strictly above or below zero, then the p-value provided is the most extreme value that can be determined by the test. In the case of highly correlated variables, the covariance matrix may be singular which will lead to k_eff being less than k (as $\rho.xy - \rho.xy.z$ would not be computed).

Value

acceleration the acceleration used for the BCa method.

alpha the proportion of the bootstrapped distribution below zero.

bias the bias used for the BCa method.

call shows the function call.

difference calculated from the data. Same as p.xy - p.xy.z.

distribution the estimated distribution of the difference as determined through bootstrapping.

k_eff the number of successful bootstrap samples. Less than or equal to k.

method the method of correlation used.

p.value significance level of the test.

p.xy Zero order correlation between x and y.

p.xy.z (semi) partial correlation between x and y while accounting for z.

semi logical. If TRUE, p.xy.z is the semi-partial correlation. Otherwise p.xy.z is the partial correlation.

test shows the type of test performed.

See Also

pzconf

Examples

```
require(graphics)
require(MASS)
# data
set.seed(1111)
mu <- rep(0,4)
Sigma <- matrix(.2, nrow=4, ncol=4) + diag(4)*.8
data <- mvrnorm(n=100, mu=mu, Sigma=Sigma)

# p.(1,2) = p.(1,2)|(3,4) test
test <- pzcor(data[,1], data[,2], data[,c(3,4)], k = 2000, semi = FALSE,
              test = 'eq')
hist(test$distribution)
test
```

Index