Package: corrfuns (via r-universe)

August 22, 2024

Title Correlation Coefficient Related Functions

Version 1.0

Date 2023-10-26

Author Michail Tsagris [aut, cre]

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Depends R (>= 4.0)

Imports graphics, Rfast, stats

Description Many correlation coefficient related functions are offered, such as correlations, partial correlations and hypothesis testing using asymptotic tests and computer intensive methods (bootstrap and permutation). References include Mardia K.V., Kent J.T. and Bibby J.M. (1979). ``Multivariate Analysis". ISBN: 978-0124712522. London: Academic Press.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-10-27 07:30:02 UTC

Contents

`corrfuns-package` *Correlation Coefficient Related Functions*

Description

Description: Many correlation coefficient related functions, such as partial correlations and hypothesis testing. The package serves an educational purpose as well, since some functions are written using a **for** loop and a **vectorised** version.

Details

Package:	corrfuns
Type:	Package
Version:	1.0
Date:	2023-10-26
License:	GPL-2

Maintainers

Michail Tsagris <mtsagris@uoc.gr>.

Author(s)

Michail Tsagris `<mtsagris@uoc.gr>`

`Asymptotic p-value for a correlation coefficient`
Asymptotic p-value for a correlation coefficient

Description

Asymptotic p-value a correlation coefficient.

Usage

```
correl(y, x, type = "pearson", rho = 0, alpha = 0.05)
```

Arguments

y	A numerical vector.
x	A numerical vector.
type	The type of correlation coefficient to compute, "pearson" or "spearman".
rho	The hypothesized value of the true partial correlation.
alpha	The significance level.

Details

Fisher's transformation for the correlation coefficient is defined as $\hat{z} = \frac{1}{2}\log\frac{1+r}{1-r}$ and its inverse is equal to $\frac{\exp(2\hat{z})-1}{\exp(2\hat{z})+1}$. The estimated standard error of Fisher's transform is $\frac{1}{\sqrt{n-3}}$ (Efron and Tibshirani, 1993, pg. 54). If on the other hand, you choose to calculate Spearman's correlation coefficients, the estimated standard error is slightly different $\simeq \frac{1.029563}{\sqrt{n-3}}$ (Fieller, Hartley and Pearson, 1957, Fieller and Pearson, 1961). R calculates confidence intervals based in a different way and does hypothesis testing for zero values only. The function calculates asymptotic confidence intervals based upon Fisher's transform, assuming asymptotic normality of the transform and performs hypothesis testing for the true (any, non only zero) value of the correlation. The sample distribution though is a t_{n-3}.

Value

A list including:

result	The correlation coefficient and the p-value for the test of zero correlation.
ci	The asymptotic $(1-\alpha)\%$ confidence interval for the true correlation coefficient.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Efron B. and Tibshirani R.J. (1993). An introduction to the bootstrap. Chapman & Hall/CRC.

Fieller E.C., Hartley H.O. and Pearson E.S. (1957). Tests for rank correlation coefficients. I. Biometrika, 44(3/4): 470–481.

Fieller E.C. and Pearson E.S. (1961). Tests for rank correlation coefficients: II. Biometrika, 48(1/2): 29–40.

See Also

correls, permcorrels

Examples

```
a <- correl( iris[, 1], iris[, 2] )
```

Asymptotic p-value for many correlation coefficients
Asymptotic p-value for many correlation coefficients

Description

Asymptotic p-value for many correlation coefficients.

Usage

```
correls(y, x, type = "pearson", rho = 0, alpha = 0.05)
```

Arguments

y	A numerical vector.
x	A numerical vector.
type	The type of correlation coefficient to compute, "pearson" or "spearman".
rho	The hypothesized value of the true partial correlation.
alpha	The significance level.

Details

Suppose you have a (dependent) variable Y and a matrix of p variables $\mathbf{X}$ and you want to get all the correlations between Y and X_i for $i = 1, \ldots, p$. if you type cor(y, x) in you will get a vector of the correlations. What I offer here is confidence interval for each of the correlations, the test statistic and the p-values for the hypothesis that each of them is equal to some value ρ. The p-values and test statistics are useful for meta-analysis for example, combination of the p-values in one or even to see the false discovery rate (see the package **fdrtool** by Korbinian Strimmer).

Value

A matrix with 5 columns, the correlations, the test statistics, their associated p-values and the relevant $(1 - \alpha)\%$ confidence intervals.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

correl, permcorrels

Examples

```
y <- rnorm(40)
x <- matrix(rnorm(40 * 1000), ncol = 1000)
a <- correls(y, x )
```

`Bootstrap p-value for the correlation coefficient`
Bootstrap p-value for the correlation coefficient

Description

Bootstrap p-value for the correlation coefficient.

Usage

```
bootcor(x, B = 999)
bootcor2(x, B = 999)
```

Arguments

`x`	A numerical matrix with two columns.
`B`	The number of bootstrap samples to generate.

Details

The functions perform non-parametric bootstrap hypothesis testing that the correlation coefficient is zero. A good pivotal statistic is the Fisher's transformation (see `correl`). Then the data have to be transformed under the null hypothesis ($\rho = 0$). This is doable via the eigen-analysis of the covariance matrix. We transform the bivariate data such that the covariance (and thus the correlation) matrix equals the identity matrix. remind that the correlation matrix is independent of measurements and is location free. The next step is easy, we draw bootstrap samples (from the transformed data) and every time we calculate the Fisher's transformation. The bootstrap p-value is calculated in the usual way (Davison and Hinkley, 1997).

If you want to perform a non-parametric bootstrap hypothesis for a value of the correlation other than zero the procedure is similar. The data have already been transformed such that their correlation is zero. Now instead of the zeroes in the off-diagonal values of the identity matrix you will have the value of the correlation matrix you want to test. Eigen analysis of the matrix is performed and the square root of the matrix is used to multiply the transformed data. I could write a more general function to include all case, but I will leave this task to you. If you do write it please send it to me and I will put it with your name of course.

The function "bootcor()" is a vectorised version of "bootcor2()". Instead of using a **for** loop you can do things vectorised. This idea cam when I found the vectorised bootstrap correlation by Neto (2015). I cannot say I understood fully what he did, so I decided to write my own code based on the direction he pointed.

Pearson's correlation coefficient of x and y for a sample size n is given by

$$r = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sqrt{\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^2\right)\left(\sum_{i=1}^{n} y_i^2 - n\bar{y}^2\right)}}$$

So, we can see that need 5 terms to calculate, $\sum_{i=1}^{n} x_i y_i, \bar{x}, \bar{y}, \sum_{i=1}^{n} x_i^2$ and $\sum_{i=1}^{n} y_i^2$. After transforming the data under the null hypothesis using the spectral decomposition we proceed as follows with B number of resamples.

Algorithm for vectorised bootstrap

1. Set a seed number in R. This is to make sure that the pairs of (x_i, y_i) are still the same.

2. Sample with replacement $B \times n$ values of x and put them in a matrix with n rows and B columns, named X_B.

3. Sample with replacement $B \times n$ values of y and put them in a matrix with n rows and B columns, names Y_B.

4. Calculate the mean vectors of X_B and Y_B.

5. Calculate the sum vector of X_B^2 and Y_B^2.

6. Finally calculate the sum vector of $X_B * Y_B$. This is the term $\sum_{i=1}^{n} x_i y_i$ for all resamples.

So we now have 5 vectors containing the 5 terms we want. We calculate the correlation coefficient and then the Fisher's transformation (see `correl`) and so we have B bootstrap test statistics. In order to see the time gain I tested both of these functions with $B = 9999$ resamples and 1000 repetitions. The gain is not super wow, I would like it if it was 1/10, but even saw, it is still good. Parallelised versions reduce time to 1/3, so from this perspective, I did better. If we now put parallel inside this vectorised version, computations will be even faster. I leave this with you.

But, I noticed one thing, the same thing Neto (2015) mentions. For big sample sizes, for example 1000 pairs, the time difference is not that big and perhaps a **for** loop is faster. The big difference is in the small to moderate sample sizes. At least for this example. What I mean by this is that you should not be afraid and say, then why? If I have big sample, I do not need vectorization. Maybe yes, but even then I still recommend it. Maybe someone else will have a better alternative for vectorization which is better even in the big samples, for the correlation of course. In the contour plots though, vectorised versions are always faster no matter what.

Value

The correlation coefficient and the bootstrap based p-value for the test of zero correlation.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Davison A.C. and Hinkley D.V. (1997). Bootstrap methods and their application. Cambridge university Press.

Neto E.C. (2015). Speeding up non-parametric bootstrap computations for statistics based on sample moments in small/moderate sample size applications. PloS ONE, 10(6): e0131333.

See Also

`permcor`

Examples

```
bootcor( iris[, 1:2] )
```

`Hypothesis test for equality of two correlation coefficients`
Hypothesis test for equality of two correlation coefficients

Description

Hypothesis test for equality of two correlation coefficients.

Usage

```
correls2.test(r1, r2, n1, n2, type = "pearson")
```

Arguments

`r1`	The value of the first correlation coefficient.
`r2`	The value of the second correlation coefficient.
`n1`	The sample size of the first sample from which the first correlation coefficient was computed.
`n2`	The sample size of the second sample from which the first correlation coefficient was computed.
`type`	The type of correlation coefficients, "pearson" or "spearman".

Details

The test statistic for the hypothesis of equality of two correlation coefficients is the following:

$$Z = \frac{\hat{z}_1 - \hat{z}_2}{\sqrt{1/\left(n1 - 3\right) + 1/\left(n2 - 3\right)}},$$

where $\hat{z}_1$ and $\hat{z}_2$ denote the Fisher's transformation (see `correl` applied to the two correlation coefficients and n_1 and n_2 denote the sample sizes of the two correlation coefficients. The denominator is the sum of the variances of the two coefficients and as you can see we used a different variance estimator than the one we used before. This function performs hypothesis testing for the equality of two correlation coefficients. The result is the calculated p-value from the standard normal distribution.

Value

The test statistic and its associated p-value for the test of equal correlations.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

correl, correls

Examples

```
y <- rnorm(40)
x <- matrix(rnorm(40 * 1000), ncol = 1000)
a <- correls(y, x )
```

Partial correlation between two variables

Partial correlation between two variables

Description

Partial correlation between two variables.

Usage

```
partialcor2(y, x, z, type = "pearson", rho = 0, alpha = 0.05)
```

Arguments

y	A numerical vector.
x	A numerical vector.
z	A numerical vector or a numerical matrix.
type	The type of partial correlation coefficient to compute, "pearson" or "spearman".
rho	The hypothesized value of the true partial correlation.
alpha	The significance level.

Details

Suppose you want to calculate the correlation coefficient between two variables controlling for the effect of (or conditioning on) one or more other variables. So you cant to calculate $\hat{\rho}\left(X, Y|\mathbf{Z}\right)$, where $\mathbf{Z}$ is a matrix, since it does not have to be just one variable. This idea was captures by Ronald Fisher some years ago. To calculate it, one can use linear regression as follows.

1. Calculate the residuals $\hat{e}_x$ from the linear regression $X = a + bZ$.

2. Calculate the residuals $\hat{e}_y$ from the linear regression $Y = c + dZ$.

3. Calculate the correlation between $\hat{e}_x$ and $\hat{e}_y$. This is the partial correlation coefficient between X and Y controlling for $\mathbf{Z}$.

The standard error of the Fisher's transformation of the sample partial correlation is Anderson (2003): $\text{SE}\left(\frac{1}{2}\log\frac{1+\hat{\rho}(X,Y|\mathbf{Z})}{1-\hat{\rho}(X,Y|\mathbf{Z})}\right) = \frac{1}{n-d-3}$, where n is the sample size and d is the number of variables upon which we control. The standard error is very similar to the one of the classical correlation coefficient. In fact, the latter one is a special case of the first when $d = 0$ and thus there is no variable whose effect is to be controlled.

Value

A list including:

`result`	The partial correlation coefficient and the p-value for the test of zero partial correlation.
`ci`	The asymptotic $(1 - \alpha)\%$ confidence interval for the true partial correlation coefficient.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

`partialcor`, `pcormat`

Examples

```
x <- iris[, 1:4]
partialcor2(x[, 1], x[, 2], x[, 3:4])
```

Partial correlation between two variables given a correlation matrix

Partial correlation between two variables when a correlation matrix is given

---

Description

Partial correlation between two variables when a correlation matrix is given.

Usage

```
partialcor(R, indx, indy, indz, n)
```

Arguments

`R`	A correlation matrix.
`indx`	The index of the first variable whose conditional correlation is to estimated.
`indy`	The index of the second variable whose conditional correlation is to estimated.
`indz`	The index of the conditioning variables.
`n`	The sample size of the data from which the correlation matrix was computed.

Details

Suppose you want to calculate the correlation coefficient between two variables controlling for the effect of (or conditioning on) one or more other variables. So you cant to calculate $\hat{\rho}\left(X, Y|\mathbf{Z}\right)$, where $\mathbf{Z}$ is a matrix, since it does not have to be just one variable. Using the correlation matrix R we can do the following:

$$r_{X,Y|\mathbf{Z}} = \begin{array}{ll} \frac{R_{X,Y} - R_{X,\mathbf{Z}} R_{Y,\mathbf{Z}}}{\sqrt{\left(1-R_{X,\mathbf{Z}}^2\right)^T \left(1-R_{Y,\mathbf{Z}}^2\right)}} & \text{if } |\mathbf{Z}| = 1 \\ -\frac{\mathbf{A}_{1,2}}{\sqrt{\mathbf{A}_{1,1}\mathbf{A}_{2,2}}} & \text{if } |\mathbf{Z}| > 1 \end{array}$$

The $R_{X,Y}$ is the correlation between variables X and Y, $R_{X,\mathbf{Z}}$ and $R_{Y,\mathbf{Z}}$ denote the correlations between X & $\mathbf{Z}$ and Y & $\mathbf{Z}$, $\mathbf{A} = \mathbf{R}_{X,Y,\mathbf{Z}}^{-1}$, with $\mathbf{A}$ denoting the correlation sub-matrix of variables $X, Y, \mathbf{Z}$ and $A_{i,j}$ denotes the element in the i-th row and j-th column of matrix A. The $|\mathbf{Z}|$ denotes the cardinality of $\mathbf{Z}$, i.e. the number of variables.

Value

The partial correlation coefficient and the p-value for the test of zero partial correlation.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

partialcor2, pcormat

Examples

```
r <- cor(iris[, 1:4])
partialcor(r, 1, 2, 3:4, 150)
```

`Partial correlation matrix`

Partial correlation matrix

Description

Partial correlation matrix.

Usage

```
pcormat(x, type = "pearson")
```

Arguments

x	A numerical matrix with two columns.
type	The type of the partial correlation matrix to compute, "pearson" or "spearman".

Details

The function computes the partial correlation matrix. Given a correlation matrix, it will return the partial correlation matrix. Each entry in the final matrix, is the partial correlation matrix between a pair of variables given all the rest variables.

Value

The partial correlation matrix.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

partialcor, partialcor2

Examples

```
pcormat( iris[, 1:4] )
```

Permutation p-value for many correlation coefficients

Permutation p-value for many correlation coefficients

Description

Permutation p-value for many correlation coefficients.

Usage

```
permcorrels(y, x, B = 999)
```

Arguments

y	A numerical vector.
x	A numerical matrix with many columns.
B	The number of bootstrap samples to generate.

Details

This is the same function as correls, only this time the p-values are produced via permutations and no confidence intervals are produced.

Value

A matrix with 2 columns, the correlations and their permutation based p-values.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

permcor, correls

Examples

```
y <- rnorm(40)
x <- matrix(rnorm(40 * 1000), ncol = 1000)
a <- permcorrels(y, x )
```

Permutation p-value for the correlation coefficient

Permutation p-value for the correlation coefficient

Description

Permutation p-value for the correlation coefficient.

Usage

```
permcor(x, B = 999, fast = TRUE)
permcor2(x, B = 999)
permcor3(x, B = 999)
```

Arguments

x	A numerical matrix with two columns.
B	The number of bootstrap samples to generate.
fast	If you want the C++ implementation leave this TRUE. It is about 2 times faster.

Details

These are permutation based p-values for the test that the true correlation coefficient is zero. The function "permcor2()" is a vectorised version of "permcor3()", whereas the function "permcor() does the following.

Instead of transforming the data under the null hypothesis and re-sampling with replacement we can permute the observations. Te basic difference is that the data are assumed to be under the null hypothesis already. Secondly, what we have to do, is to destroy the pairs. For example, the pairs (a, b), (c, d) and (e, f) in one permutation they can be (c, b), (a, f) and (e, d). And this thing will happen many times, say $B = 999$. Then we have B pseudo-samples again. Everything else is the same as in the bootstrap case. A trick is that we need not change the order of both variables, just the one is enough. This will sped up the process. And guess what, it is faster than bootstrap. It does not require the data to be transformed under the null hypothesis and you only need to permute one variable, in contrast to the bootstrap case, where you must resample from both variables. See Chatzipantsiou et al. (2019) for more details.

Value

The correlation coefficient and the permutation based p-value for the test of zero correlation.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Chatzipantsiou C., Dimitriadis M., Papadakis M. and Tsagris M. (2019). Extremely efficient permutation and bootstrap hypothesis tests using R. Journal of Modern Applied Statistical Methods, 18(2): eP2898.

See Also

bootcor

Examples

```
permcor( iris[, 1:2] )
```

`Squared multivariate correlation between two sets of variables`

Squared multivariate correlation between two sets of variables

---

Description

Squared multivariate correlation between two sets of variables.

Usage

```
sq.correl(y, x)
```

Arguments

y	A numerical matrix.
x	A numerical matrix.

Details

Mardia, Kent and Bibby (1979, pg. 171) defined two squared multiple correlation coefficient between the dependent variable $\mathbf{Y}$ and the independent variable $\mathbf{X}$. They mention that these are a similar measure of the coefficient determination in the univariate regression. Assume that the multivariate regression model is written as $\mathbf{Y} = \mathbf{XB} + \mathbf{U}$, where $\mathbf{U}$ is the matrix of residuals. Then, they write $\mathbf{D} = \left(\mathbf{Y}^T\mathbf{Y}\right)^{-1}\hat{\mathbf{U}}^T\hat{\mathbf{U}}$, with $\hat{\mathbf{U}}^T\hat{\mathbf{U}} = \mathbf{Y}^T\mathbf{PY}$ and $\mathbf{P}$ is $\mathbf{P} = \mathbf{I}_n - \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$. The matrix $\mathbf{D}$ is a generalization of $1 - R^2$ in the univariate case. Mardia, Kent and Bibby (1979, pg. 171) mentioned that the dependent variable $\mathbf{Y}$ has to be centred.

The squared multivariate correlation should lie between 0 and 1 and this property is satisfied by the trace correlation r_T and the determinant correlation r_D, defined as $r_T^2 = d^{-1}\text{tr}\left(\mathbf{I} - \mathbf{D}\right)$ and $r_D^2 = \det\left(\mathbf{I} - \mathbf{D}\right)$ respectively, where d denotes the dimensionality of $\mathbf{Y}$. So, high values indicate high proportion of variance of the dependent variables explained. Alternatively, one can calculate the trace and the determinant of the matrix $\mathbf{E} = \left(\mathbf{Y}^T\mathbf{Y}\right)^{-1}\hat{\mathbf{Y}}^T\hat{\mathbf{Y}}$. Try something else also, use the function "sq.correl()" in a univariate regression example and then calculate the R^2 for the same dataset. Try this example again but without centering the dependent variable. In addition, take two variables and calculate their squared correlation coefficient and then square it and using "sq.correl()".

Value

A vector with two values, the trace and determinant R^2.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

correl

Examples

```
sq.correl( iris[, 1:2], iris[, 3:4] )
```

Index