Package 'Rfast2'

December 16, 2019

Type Package

Title A Collection of Efficient and Extremely Fast R Functions II

Version 0.0.5

Date 2019-12-16

Author Manos Papadakis, Michail Tsagris, Stefanos Fafalios and Marios Dimitriadis.

Maintainer Manos Papadakis <rfastofficial@gmail.com>

Depends R (>= 3.5.0), Rcpp (>= 0.12.3)

LinkingTo Rcpp (>= 0.12.3), RcppArmadillo

Imports Rfast

SystemRequirements C++11

BugReports https://github.com/RfastOfficial/Rfast2/issues

URL https://github.com/RfastOfficial/Rfast2

Description A collection of fast statistical and utility functions for data analysis. Functions for regression, maximum likelihood, column-wise statistics and many more have been included. C++ has been utilized to speed up the functions.

LazyData TRUE

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-12-16 16:00:16 UTC

R topics documented:

`Rfast2-package` *Really fast R functions*

Description

A collection of Rfast2 functions for data analysis. Note 1: The vast majority of the functions accept matrices only, not data.frames. Note 2: Do not have matrices or vectors with have missing data (i.e NAs). We do no check about them and C++ internally transforms them into zeros (0), so you may get wrong results. Note 3: In general, make sure you give the correct input, in order to get the correct output. We do no checks and this is one of the many reasons we are fast.

Details

Package:	Rfast2
Type:	Package
Version:	0.0.5
Date:	2019-12-16
License:	GPL-2

Maintainers

Manos Papadakis <rfastofficial@gmail.com>

Author(s)

Manos Papadakis <papadakm95@gmail.com>, Michail Tsagris <mtsagris@yahoo.gr>, Stefanos Fafalios <stefanosfafalios@gmail.com>, Marios Dimitriadis <kmdimitriadis@gmail.com>.

Add many single terms to a model

Add many single terms to a model

Description

Add many single terms to a model.

Usage

```
add.term(y, xinc, xout, devi_0, type = "logistic", logged = FALSE,
tol = 1e-07, maxiters = 100, parallel = FALSE)
```

Arguments

y	The response variable. It must be a numerical vector.
xinc	The already included indendent variable(s).
xout	The independent variables whose conditional association with the response is to be calculated.
devi_0	The deviance for Poisson, logistic, qpoisson, qlogistic and normlog regression or the log-likelihood for the Weibull, spml and multinomial regressions. See the example to understand better.
type	The type of regression, "poisson", "logistic", "qpoisson" (quasi Poisson), "qlogistic" (quasi logistic) "normlog" (Gaussian regression with log-link) "weibull", "spml" and "multinom".
logged	Should the logarithm of the p-value be returned? TRUE or FALSE.
tol	The tolerance value to terminate the Newton-Raphson algorithm when fitting the regression models.
maxiters	The maximum number of iterations the Newton-Raphson algorithm will perform.
parallel	Should the computations take place in parallel? TRUE or FALSE.

Details

The function is similar to the built-in function add1. You have already fitted a regression model with some independent variables (xinc). You then add each of the xout variables and test their significance.

Value

A matrix with two columns. The test statistic and its associated (logged) p-value.

Author(s)

Stefanos Fafalios

R implementation and documentation: Stefanos Fafalios <stefanosfafalios@gmail.com>.

References

McCullagh, Peter, and John A. Nelder. Generalized linear models. CRC press, USA, 2nd edition, 1989.

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

See Also

bic.regs, logiquant.regs, sp.logiregs

Examples

```
x <- matrix( rnorm(200 * 10), ncol = 10)
y <- rpois(200, 10)
devi_0 <- deviance( glm(y ~ x[, 1:2], poisson) )
a <- add.term(y, xinc = x[,1:2], xout = x[, 3:10], devi_0 = devi_0, type= "poisson")

y <- rbinom(200, 1, 0.5)
devi_0 <- deviance( glm(y ~ x[, 1:2], binomial) )
a <- add.term(y, xinc = x[,1:2], xout = x[, 3:10], devi_0 = devi_0, type= "logistic")


y <- rbinom(200, 2, 0.5)
devi_0 <- Rfast::multinom.reg(y, x[, 1:2])$loglik
a <- add.term(y, xinc = x[,1:2], xout = x[, 3:10], devi_0 = devi_0, type= "multinom")

y <- rgamma(200, 3, 1)
devi_0 <- Rfast::weib.reg(y, x[, 1:2])$loglik
a <- add.term(y, xinc = x[,1:2], xout = x[, 3:10], devi_0 = devi_0, type= "weibull")
```

`Angular Gaussian random values simulation`

Angular Gaussian random values simulation

Description

Angular Gaussian random values simulation.

Usage

```
riag(n, mu)
```

Arguments

n	The sample size, a numerical value.
mu	The mean vector in R^d.

Details

The algorithm uses univariate normal random values and with some mean. The vectors are then scaled to have unit length.

Value

A matrix with the simulated data.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

Paine P.J., Preston S.P., Tsagris M and Wood A.T.A. (2018). An Elliptically Symmetric Angular Gaussian Distribution. Statistics and Computing, 28(3):689–697.

See Also

colspml.mle, circ.cor1, circ.cors1

Examples

```
x <- riag(20, rnorm(4, 3, 1))
```

Anova for circular data

Analysis of variance for circular data

Description

Analysis of variance for circular data.

Usage

```
hcf.circaov(u, ina)

lr.circaov(u, ina)

het.circaov(u, ina)

embed.circaov(u, ina)
```

Arguments

u	A numeric vector containing the data that are expressed in rads.
ina	A numerical or factor variable indicating the group of each value.

Details

The high concentration (hcf.circaov), log-likelihood ratio (lr.circaov), embedding approach (embed.circaov) or the non equal concentration parameters approach (het.circaov) is used.

Value

A vector including:

test	The value of the test statistic.
p-value	The p-value of the test.
kapa	The concentration parameter based on all the data. If the het.circaov is used this argument is not returned.

Author(s)

Michail Tsagris

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

References

Mardia, K. V. and Jupp, P. E. (2000). Directional statistics. Chicester: John Wiley & Sons.

See Also

multivm.mle, vm.nb

Examples

```
x <- rnorm(60, 2.3, 0.3)
ina <- rep(1:3,each = 20)
hcf.circaov(x, ina)
lr.circaov(x, ina)
het.circaov(x, ina)
embed.circaov(x, ina)
```

`Benchmark - Measure time`

Benchmark - Measure time

Description

Lower/upper triangular matrix.

Usage

```
benchmark(...,times,envir=parent.frame(),order=NULL)
## S3 method for class 'benchmark'
print(x,...)
```

Arguments

`...`	Expressions to the benchmark function.
`x`	Object of class "benchmark" to print.
`times`	Number of time to measure execution time of the expression.
`envir`	Environment to evaluate the expressions.
`order`	An integer vector to execute the epxressions with this order, otherwise the execution order is random.

Details

For measuring time we have used C++'s new library "chrono".

Value

The execution time for each expression.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

See Also

`Quantile`, `trim.mean`

Examples

```
benchmark(x <- matrix(runif(10*10),10,10),times=10)
```

BIC of many simple univariate regressions
BIC of many simple univariate regressions.

Description

BIC of many simple univariate regressions.

Usage

```
bic.regs(y, x, family = "normal")
```

Arguments

y	The dependent variable, a numerical vector.
x	A matrix with the indendent variables.
family	The family of the regression models. "normal", "binomial", "poisson", "multinomial", "normlog" (Gaussian regression with log link), "spmpl" (SPML regression) or "weibull" for Weibull regression.

Details

Many simple univariate regressions are fitted and the BIC of every model is computed.

Value

A vector with the BIC of each regression model.

Author(s)

Michail Tsagris

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

See Also

logistic_only,poisson_only

Examples

```
y <- rbinom(100, 1, 0.6)
x <- matrix( rnorm(100 * 50), ncol = 50 )
bic.regs(y, x, "binomial")
```

Bootstrap James and Hotelling test for 2 independent sample mean vectors
Bootstrap James and Hotelling test for 2 independent sample mean vectors

---

Description

Bootstrap James and Hotelling test for 2 independent sample mean vectors.

Usage

```
boot.james(y1, y2, R = 999)
boot.hotel2(y1, y2, R = 999)
```

Arguments

y1	A numerical matrix with the data of the one sample.
y2	A numerical matrix with the data of the other sample.
R	The number of bootstrap samples to use.

Details

We bootstrap the 2-samples James (does not assume equal covariance matrics) and Hotelling test (assumes equal covariance matrics). The difference is that the Hotelling test statistic assumes equaility of the covariance matrices, which if violated leads to inlfated type I errors. Bootstrap calibration though takes care of this issue. As for the bootstrap calibration, instead of sampling B times from each sample, we sample $sqrtB$ from each of them and then take all pairs. Each bootstrap sample is independent of each other, hence there is no violation of the theory (Chatzipantsiou et al., 2019).

Value

The bootstrap p-value.

Author(s)

Michail Tsagris

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

References

G.S. James (1954). Tests of Linear Hypothese in Univariate and Multivariate Analysis when the Ratios of the Population Variances are Unknown. Biometrika, 41(1/2): 19-43

Efron Bradley and Robert J. Tibshirani (1993). An introduction to the bootstrap. New York: Chapman \& Hall/CRC.

Chatzipantsiou C., Dimitriadis M., Papadakis M. and Tsagris M. (2019). Extremely efficient permutation and bootstrap hypothesis tests using R. To appear in the Journal of Modern Applied Statistical Methods.

https://arxiv.org/ftp/arxiv/papers/1806/1806.10947.pdf

See Also

`welch.tests`, `trim.mean`

Examples

```
boot.james( as.matrix(iris[1:25, 1:4]), as.matrix(iris[26:50, 1:4]) )
```

`Bootstrap Student's t-test for 2 independent samples`
Bootstrap Student's t-test for 2 independent samples

Description

Bootstrap Student's t-test for 2 independent samples.

Usage

```
boot.student2(x, y, B = 999)
```

Arguments

`x`	A numerical vector with the data.
`y`	A numerical vector with the data.
`B`	The number of bootstrap samples to use.

Details

We bootstrap Student's (Gosset's) t-test statistic and not the Welch t-test statistic. For the latter case see the "boot.ttest2" function in Rfast. The difference is that Gosset's test statistic assumes equaility of the variances, which if violated leads to inlfated type I errors. Bootstrap calibration though takes care of this issue. As for the bootstrap calibration, instead of sampling B times from each sample, we sample $sqrt B$ from each of them and then take all pairs. Each bootstrap sample is independent of each other, hence there is no violation of the theory (Chatzipantsiou et al., 2019).

Value

A vector with the test statistic and the bootstrap p-value.

Author(s)

Michail Tsagris

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

References

Efron Bradley and Robert J. Tibshirani (1993). An introduction to the bootstrap. New York: Chapman \& Hall/CRC.

Chatzipantsiou C., Dimitriadis M., Papadakis M. and Tsagris M. (2019). Extremely efficient permutation and bootstrap hypothesis tests using R. To appear in the Journal of Modern Applied Statistical Methods.

https://arxiv.org/ftp/arxiv/papers/1806/1806.10947.pdf

See Also

welch.tests, trim.mean

Examples

```
x <- rexp(40, 4)
y <- rbeta(50, 2.5, 7.5)
system.time(t.test(x, y, var.equal = TRUE) )
system.time( a <- boot.student2(x, y, 9999) )
a
```

Check if a matrix is Lower or Upper triangular

Check if a matrix is Lower or Upper triangular

Description

Lower/upper triangular matrix.

Usage

```
is.lower.tri(x, diag = FALSE)
is.upper.tri(x, diag = FALSE)
```

Arguments

x	A matrix with data.
diag	A logical value include the diagonal to the result.

Value

Check if a matrix is lower or upper triangular. You can also include diagonal to the check.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

See Also

Intersect

Examples

```
x <- matrix(runif(10*10),10,10)

is.lower.tri(x)
is.lower.tri(x,TRUE)


is.upper.tri(x)
is.upper.tri(x,TRUE)
```

Check whether a square matrix is skew-symmetric

Check whether a square matrix is skew-symmetric

Description

Check whether a square matrix is skew-symmetric.

Usage

```
is.skew.symmetric(x)
```

Arguments

x	A square matrix with data.

Details

Instead of going through the whole matrix, the function will stop if the first disagreement is met.

Value

A boolean value, TRUE of FALSE.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

See Also

cholesky, cora, cova

Examples

```
x <-matrix( rnorm( 100 * 400), ncol = 400 )
s1 <- cor(x)
is.skew.symmetric(s1)
x <- x[1:100, ]
is.skew.symmetric(x)

x<-s1<-NULL
```

Circurlar correlations between two circular variables

Circurlar correlations between two circular variables

Description

Circurlar correlations between two circular variables.

Usage

```
circ.cor1(theta, phi, pvalue = FALSE)

circ.cors1(theta, phi, pvalue = FALSE)
```

Arguments

theta	The first cirular variable expressed in radians, not degrees.
phi	The other cirular variable. In the case of "circ.cors1" this is a matrix with many circular variables. In either case, the values must be in radians, not degrees.
pvalue	If you want the p-value of the zero correlation hypothesis testing set this to TRUE, otherwise leave it FALSE.

Details

Correlation for circular variables using the cosinus and sinus formula of Jammaladaka and Sen-Gupta (1988).

Value

If you set pvalue = TRUE, then for the "circ.cor1" a vector with two values, the correlation and its associated p-value, otherwise the correlation only. For the "circ.cors1", either a vector with the correlations only or a matrix with two columns, the correlation and the p-values.

Author(s)

Michail Tsagris

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

References

Jammalamadaka, R. S. and Sengupta, A. (2001). Topics in circular statistics. World Scientific.

Jammalamadaka, S. R. and Sarma, Y. R. (1988) . A correlation coefficient for angular variables. Statistical Theory and Data Analysis, 2:349–364.

See Also

`spml.reg`

Examples

```
y <- runif(50, 0, 2 * pi)
x <- runif(50, 0, 2 * pi)
circ.cor1(y, x, TRUE)
x <- matrix(runif(50 * 10, 0, 2 * pi), ncol = 10)
circ.cors1(y, x, TRUE)
```

`Column and row-wise jackknife sample means` *Column and row-wise jackknife sample means*

Description

Column and row-wise jackknife sample means.

Usage

```
coljack.means(x)
rowjack.means(x)
```

Arguments

`x`	A numerical matrix with data.

Details

An efficient implementation of the jackknife mean is provided.

Value

A vector with the jackknife sample means.

Author(s)

Michail Tsagris

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

References

Efron Bradley and Robert J. Tibshirani (1993). An introduction to the bootstrap. New York: Chapman \& Hall/CRC.

See Also

welch.tests, trim.mean

Examples

```
x <- as.matrix(iris[1:50, 1:4])
coljack.means(x)
```

Column-wise means and variances

Column-wise means and variances of a matrix

Description

Column-wise means and variances of a matrix.

Usage

```
colmeansvars(x, std = FALSE, parallel = FALSE)
```

Arguments

x	A matrix with the data.
std	A boolean variable specyfying whether you want the variances (FALSE) or the standard deviations (TRUE) of each column.
parallel	A boolean value for parallel version.

Details

This function cacluates the column-wise means and variances (or standard deviations).

Value

A matrix with two rows. The first contains the means and the second contains the variances (or standard deviations).

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Manos Papadakis <papadakm95@gmail.com>.

See Also

pooled.colVars

Examples

```
colmeansvars( as.matrix(iris[, 1:4]) )
```

Column-wise MLE of some univariate distributions

Column-wise MLE of some univariate distributions

Description

Column-wise MLE of some univariate distributions.

Usage

```
collognorm.mle(x)
collogitnorm.mle(x)
colborel.mle(x)
colhalfnorm.mle(x)
colordinal.mle(x, link = "logit")
```

Arguments

`x`	A numerical matrix with data. Each column refers to a different vector of observations of the same distribution. The values of for Lognormal must be greater than zero, for the logitnormal they must by percentages, exluding 0 and 1, whereas for the Borel distribution the x must contain integer values greater than 1. For the halfnormal the numbers must be strictly positive, while for the ordinal this can be a numerical matrix with values 1, 2, 3,..., not zeros.
`link`	This can either be "logit" or "probit". It is the link function to be used.

Details

For each column, the same distribution is fitted and its parameters and log-likelihood are computed.

Value

A matrix with two or three columns. The first one or the first two contain the parameter(s) of the distribution and the second or third column the relevant log-likelihood. For the ordinal a list including:

`param`	A matrix with the intercepts (threshold coefficients) of the model applied to each column (or variable).
`loglik`	The log-likelihood values.

Author(s)

Michail Tsagris

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

References

N.L. Johnson, S. Kotz \& N. Balakrishnan (1994). Continuous Univariate Distributions, Volume 1 (2nd Edition).

N.L. Johnson, S. Kotz \& N. Balakrishnan (1970). Distributions in statistics: continuous univariate distributions, Volume 2.

Agresti, A. (2002) Categorical Data. Second edition. Wiley.

See Also

censpois.mle,gammapois.mle

Examples

```
x <- matrix( exp( rnorm(1000 * 50) ), ncol = 50)
a <- collognorm.mle(x)
x <- NULL
```

Column-wise MLE of the angular Gaussian distribution

Column-wise MLE of the angular Gaussian distribution

Description

Column-wise MLE of the angular Gaussian distribution.

Usage

```
colspml.mle(x ,tol = 1e-07, maxiters = 100, parallel = FALSE)
```

Arguments

x	A numerical matrix with data. Each column refers to a different vector of observations of the same distribution. The values of for Lognormal must be greater than zero, for the logitnormal they must by percentages, exluding 0 and 1, whereas for the Borel distribution the x must contain integer values greater than 1.
tol	The tolerance value to terminate the Newton-Raphson algorithm.
maxiters	The maximum number of iterations that can take place in each regression.
parallel	Do you want this to be executed in parallel or not. The parallel takes place in C++, and the number of threads is defined by each system's availiable cores.

Details

For each column, spml.mle function is applied that fits the angular Gaussian distribution estimates its parameters and computes the maximum log-likelihood.

Value

A matrix with four columns. The first two are the mean vector, then the γ parameter, and the fourth column contains maximum log-likelihood.

Author(s)

Stefanos Fafalios

R implementation and documentation: Stefanos Fafalios <stefanosfafalios@gmail.com>

References

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

See Also

collognorm.mle,gammapois.mle

Examples

```
x <- matrix( runif(100 * 10), ncol = 10)
a <- colspml.mle(x)
x <- NULL
```

Column-wise pooled variances across groups

Column-wise pooled variances across groups

Description

Column-wise pooled variances across groups.

Usage

```
pooled.colVars(x, ina, std = FALSE)
```

Arguments

x	A matrix with the data.
ina	A numerical vector specifying the groups. If you have numerical values, do not put zeros, but 1, 2, 3 and so on.
std	A boolean variable specyfying whether you want the variances (FALSE) or the standard deviations (TRUE) of each column.

Details

This function cacluates the pooled variance (or standard deviation) for a range of groups for each column.

Value

A vector with the pooled column variances or standard deviations.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Manos Papadakis <papadakm95@gmail.com>.

See Also

`colmeansvars`

Examples

```
pooled.colVars( as.matrix(iris[, 1:4]), as.numeric(iris[, 5]) )
```

`Column-wise summary statistics with grouping variables`
Column-wise summary statistics with grouping variables

Description

Column-wise summary statistics with grouping variables.

Usage

```
colGroup(x,ina,method="sum",names=TRUE, std = FALSE)
```

Arguments

`x`	A matrix with data.
`ina`	A numerical vector specifying the groups. If you have numerical values, do not put zeros, but 1, 2, 3 and so on. **The numbers must be consecutive**, like 1,2,3,.. Do not put 1, 3, 4 as this will cause C++ to crash.
`method`	One of the: "sum", "min", "max", "median", "var".
`names`	Set the name of the result vector with the unique numbers of group variable.
`std`	A boolean variable specyfying whether you want the variances (FALSE) or the standard deviations (TRUE) of each column. This is taken into account only when method = "var".

Value

Column wise of grouping variables. You can also include diagonal to the check.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

See Also

Quantile,colQuantile,rowQuantile

Examples

```
x <- matrix(runif(100 * 5), 100, 5)
group <- sample(1:3, 100, TRUE)

all.equal( colGroup(x, group), rowsum(x, group) )
```

`Constrained least squares` *Constrained least squares*

Description

Constrained least squares.

Usage

```
cls(y, x, R, ca)
```

Arguments

y	The response variables, a numerical vector with observations.
x	A matrix with independent variables, the design matrix.
R	The R vector that contains the values that will multiply the beta coefficients. See details and examples.
ca	The value of the constraint, $R^T\beta = c$. See details and examples.

Details

This is described in Chapter 8.2 of Hansen (2019). The idea is to inimise the sum of squares of the residuals under the constraint $R^T\beta = c$. As mentioned above, be careful with the input you give in the x matrix and the R vector.

Value

A list including:

bols	The OLS (Ordinary Least Squares) beta coefficients.
bcls	The CLS (Constrained Least Squares) beta coefficients.

Author(s)

Michail Tsagris

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

References

Hansen, B. E. (2019). Econometrics. https://www.ssc.wisc.edu/~bhansen/econometrics/Econometrics.pdf

See Also

gee.reg, bic.regs, ztp.reg

Examples

```
x <- as.matrix( iris[1:50, 1:4] )
y <- rnorm(50)
R <- c(1, 1, 1, 1)
cls(y, x, R, 1)
```

`Correlation significance testing using Fisher's z-transformation`
Correlation significance testing using Fisher's z-transformation

---

Description

Correlation significance testing using Fisher's z-transformation.

Usage

```
cor_test(y, x, type = "pearson", rho = 0, a = 0.05 )
```

Arguments

y	A numerical vector.
x	A numerical vector.
type	The type of correlation you want. "pearson" and "spearman" are the two supported types because their standard error is easily calculated.
rho	The value of the hypothesised correlation to be used in the hypothesis testing.
a	The significance level used for the confidence intervals.

Details

The function uses the built-in function "cor" which is very fast, then computes a confidence interval and produces a p-value for the hypothesis test.

Value

A vector with 5 numbers; the correlation, the p-value for the hypothesis test that each of them is equal to "rho", the test statistic and the $a/2%$ lower and upper confidence limits.

Author(s)

Michail Tsagris

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

See Also

allbetas,univglms

Examples

```
x <- rcauchy(60)
y <- rnorm(60)
cor_test(y, x)
```

Covariance between a variable and a matrix of variables

Covariance between a variable and a matrix of variables

Description

Covariance between a variable and a matrix of variables.

Usage

```
covar(y, x)
```

Arguments

y	A numerical vector.
x	A numerical matrix.

Details

The function calculates the covariance between a variable and many others.

Value

A vector with the covariances.

Author(s)

Michail Tsagris and Manos Papadakis

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

See Also

circ.cors1,bic.regs

Examples

```
y <- rnorm(40)
x <- matrix( rnorm(40 * 10), ncol = 10 )
covar(y, x)
cov(y, x)
```

Diagonal values of the Hat matrix
Diagonal values of the Hat matrix

Description

Diagonal values of the Hat matrix.

Usage

```
leverage(x)
```

Arguments

x A matrix with independent variables, the design matrix.

Details

The function returns the diagonal values of the Hat matrix used in linear regression. We did not call it "hatvalues" as R contains a built-in function with such a name.

Value

A vector with the diagonal Hat matrix values, the leverage of each observation.

Author(s)

Michail Tsagris

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

References

Hansen, B. E. (2019). Econometrics. https://www.ssc.wisc.edu/~bhansen/econometrics/Econometrics.pdf

See Also

gee.reg, bic.regs, ztp.reg

Examples

```
x <- as.matrix( iris[1:50, 1:4] )
a <- leverage(x)
```

Empirical entropy *Empirical entropy*

Description

Empirical entropy.

Usage

```
empirical.entropy(x, k = NULL, pretty = FALSE)
```

Arguments

x	A numerical vector with continuous values.
k	If you want to cut the data into a specific range plug it here, otherwise this decide based upon the Freedman-Diaconis' rule.
pretty	Should the breaks be equally space upon the range of x? If yes, let this FALSE. If this is TRUE, the breaks are decided using the base command pretty.

Details

The function computes the empirical entropy.

Value

The estimated empirical entropy.

Author(s)

Michail Tsagris

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

References

https://en.wikipedia.org/wiki/Entropy_estimation

https://en.wikipedia.org/wiki/Histogram

Freedman David and Diaconis P. (1981). On the histogram as a density estimator: L2 theory. Zeitschrift fur Wahrscheinlichkeitstheorie und Verwandte Gebiete. 57(4): 453-476.

See Also

Quantile, pretty

Examples

```
x <- rnorm(100)
empirical.entropy(x)
empirical.entropy(x, pretty = TRUE)
```

Fixed intercepts Poisson regression

Fixed intercepts Poisson regression

Description

Fixed intercepts Poisson regression.

Usage

```
fipois.reg(y, x, id, tol = 1e-07, maxiters = 100)
```

Arguments

y	The dependent variable, a numerical vector with integer, non negative valued data.
x	A matrix with the indendent variables.
id	A numerical variable with 1, 2, ... indicating the subject. Unbalanced design is of course welcome.
tol	The tolerance value to terminate the Newton-Raphson algorithm. This is set to 10^{-7} by default.
maxiters	The maximum number of iterations that can take place during the fitting.

Details

Fixed intercepts Poisson regression for clustered count data is fitted. According to Demidenko (2013), when the number of clusters (N) is small and the number of observations per cluster (n_i) is relatively large, say $min(n_i) > N$, one may assume that the intercept $\alpha_i = \beta + u_i$ is fixed and unknown ($i = 1, ..., N$).

Value

A list including:

be	The regression coefficients.
seb	The standard errors of the regression coefficients.
ai	The estimated fixed intercepts fore ach cluster of observations.
covbeta	The covariance matrix of the regression coefficients.
loglik	The maximised log-likelihood value.
iters	The number of iteration the Newton-Raphson required.

Author(s)

Michail Tsagris

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

References

Eugene Demidenko (2013). Mixed Models: Theory and Applications with R, pages 388-389, 2nd Edition. New Jersey: Wiley \& Sons (excellent book).

See Also

cluster.lm, covar, welch.tests

Examples

```
y <- rpois(200, 10)
id <- sample(1:10, 200, replace = TRUE)
x <- rpois(200, 10)
fipois.reg(y, x, id)
```

Forward Backward Early Dropping selection regression

Forward Backward Early Dropping selection regression

Description

Forward Backward Early Dropping selection regression.

Usage

```
fbed.reg(y, x, alpha = 0.05, type = "logistic", K = 0, backward = FALSE,
         parallel = FALSE, tol = 1e-07, maxiters = 100)
```

Arguments

y	The response variable, a numeric vector.
x	A matrix with continuous variables.
alpha	The significance threshold value for assessing p-values. Default value is 0.05.
type	The available types are: "logistic" (binary logistic regression), "qlogistic" (quasi logistic regression, for binary value or proportions including 0 and 1), "poisson" (Poisson regression), "qpoisson" (quasi Poisson regression), "weibull" (Weibull regression) and "spml" (SPML regression).
K	How many times should the process be repeated? The default value is 0.
backward	After the Forward Early Dropping phase, the algorithm proceeds witha the usual Backward Selection phase. The default value is set to TRUE. It is advised to perform this step as maybe some variables are false positives, they were wrongly selected. This is rather experimental now and there could be some mistakes in the indices of the selected variables. Do not use it for now.
parallel	If you want the algorithm to run in parallel set this TRUE.
tol	The tolerance value to terminate the Newton-Raphson algorithm.
maxiters	The maximum number of iterations Newton-Raphson will perform.

Details

The algorithm is a variation of the usual forward selection. At every step, the most significant variable enters the selected variables set. In addition, only the significant variables stay and are further examined. The non signifcant ones are dropped. This goes until no variable can enter the set. The user has the option to re-do this step 1 or more times (the argument K). In the end, a backward selection is performed to remove falsely selected variables. Note that you may have specified, for example, K=10, but the maximum value FBED used can be 4 for example.

The "qlogistic" and "qpoisson" proceed with the Wald test and no backward is performed, while for all the other regression types, the log-likelihood ratio test is used and backward phase is available.

Value

If K is a single number a list including:

Note, that the "gam" argument must be the same though.

res	A matrix with the selected variables and their test statistic.
info	A matrix with the number of variables and the number of tests performed (or models fitted) at each round (value of K). This refers to the forward phase only.
runtime	The runtime required.

Author(s)

Michail Tsagris and Stefanos Fafalios

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Stefanos Fafalios <stefanosfafalios@gmail.com>

References

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

Tsagis M. (2018). Guide on performing feature selection with the R package MXM. http://mensxmachina.org/site/wp-content/uploads/2018/04/Guide-on-performing-feature-selection-with-the-R-pdf

See Also

logiquant.regs, bic.regs, gee.reg

Examples

```
#simulate a dataset with continuous data
x <- matrix( runif(100 * 50, 1, 100), ncol = 50 )
y <- rnbinom(100, 10, 0.5)
a <- fbed.reg(y, x, type = "poisson")
```

`Gamma regression with a log-link`

Gamma regression with a log-link

Description

Gamma regression with a log-link.

Usage

```
gammareg(y, x, tol = 1e-07, maxiters = 100)
```

Arguments

`y`	The dependent variable, a numerical variable with non negative numbers.
`x`	A matrix or data.frame with the indendent variables.
`tol`	The tolerance value to terminate the Newton-Raphson algorithm.
`maxiters`	The maximum number of iterations that can take place in the regression.

Details

The gamma.reg fits a Gamma regression with a log-link. The gamma.con fits a Gamma regression with a log link with the intercept only (glm(y ~ 1, Gamma(log))).

Value

A list including:

iters	The number of iterations required by the newton-Raphson.
deviance	The deviance value.
phi	The dispersion parameter (ϕ) of the regression. This is necessary if you want to perform an F hypothesis test for the significance of one or more independent variables.
be	The regression coefficient(s).

Author(s)

Michail Tsagris

R implementation and documentation: Stefanos Fafalios <stefanosfafalios@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>

References

McCullagh, Peter, and John A. Nelder. Generalized linear models. CRC press, USA, 2nd edition, 1989.

See Also

gammaregs, zigamma.mle

Examples

```
y <- rgamma(100, 3, 4)
x <- matrix( rnorm(100 * 2), ncol = 2)
m1 <- glm(y ~ x, family = Gamma(log) )
m2 <- gammareg(y, x)
```

GEE Gaussian regression

GEE Gaussian regression

Description

GEE Gaussian regression.

Usage

```
gee.reg(y, x, id, tol = 1e-07, maxiters = 100)
```

Arguments

y	The dependent variable, a numerical vector.
x	A matrix with the indendent variables.
id	A numerical variable with 1, 2, ... indicating the subject. Unbalanced design is of course welcome.
tol	The tolerance value to terminate the Newton-Raphson algorithm. This is set to 10^{-7} by default.
maxiters	The maximum number of iterations that can take place during the fitting.

Details

Gaussin GEE regression is fitted.

Value

A list including:

be	The regression coefficients.
seb	The standard errors of the regression coefficients.
phi	The ϕ parameter.
a	The α parameter.
covbeta	The covariance matrix of the regression coefficients.
iters	The number of iteration the Newton-Raphson required.

Author(s)

Michail Tsagris

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

References

Wang M. (2014). Generalized estimating equations in longitudinal data analysis: a review and recent developments. Advances in Statistics, 2014.

Hardin J. W. and Hilbe J. M. (2002). Generalized estimating equations. Chapman and Hall/CRC.

See Also

cluster.lm, fipois.reg, covar, welch.tests

Examples

```
y <- rnorm(200)
id <- sample(1:20, 200, replace = TRUE)
x <- rnorm(200, 3)
gee.reg(y, x, id)
```

Gumbel regression *Gumbel regression*

Description

Gumbel regression.

Usage

```
gumbel.reg(y, x, tol = 1e-07, maxiters = 100)
```

Arguments

y	The dependent variable, a numerical vector with real valued numbers.
x	A matrix or a data.frame with the indendent variables.
tol	The tolerance value required by the Newton-Raphson to stop.
maxiters	The maximum iterations allowed.

Details

A Gumbel regression model is fitted. the standard errors of the regressions are not returned as we do not compute the full Hessian matrix at each step of the Newton-Raphson.

Value

A list including:

be	The regression coefficients.
sigma	The scale parameter.
loglik	The loglikelihood of the regression model.
iters	The iterations required by the Newton-Raphson.

Author(s)

Michail Tsagris

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

See Also

negbin.reg, ztp.reg

Examples

```
y <- rnorm(100)
x <- matrix(rnorm(100 * 3), ncol = 3)
mod <- gumbel.reg(y, x)
```

Intersect *Intersect Operation*

Description

Performs intersection in the same manner as R's base package intersect works.

Usage

```
Intersect(x, y)
```

Arguments

x, y	vectors containing a sequence of items, ideally of the same mode

Details

The function will discard any duplicated values in the arguments.

Value

The function will return a vector of the same mode as the arguments given. NAs will be removed.

Author(s)

Marios Dimitriadis

R implementation and documentation: Marios Dimitriadis <kmdimitriadis@gmail.com>

See Also

intersect

Examples

```
x <- c(sort(sample(1:20, 9)))
y <- c(sort(sample(3, 23, 7)))
Intersect(x, y)
```

`Item difficulty and discrimination` *Item difficulty and discrimination*

Description

Item difficulty and discrimination.

Usage

```
diffic(x)

discrim(x, frac = 1/3)
```

Arguments

`x`	A numerical matrix with 0s (wrong answer) and 1s (correct answer).
`frac`	A number between 0 and 1 used to calculate the difficulty of each question.

Details

The difficulty and the discrimination of each question (item) are calculated.

Value

A vector with the item difficulties or item discriminations.

Author(s)

Michail Tsagris

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

References

Kaplan E. L. and Meier P. (1958). Nonparametric estimation from incomplete observations. Journal of the American Statistical Association, 53(282): 457-481.

See Also

`Quantile`, `colmeansvars`

Examples

```
x <- matrix(rbinom(100 * 10, 1, 0.7), ncol = 10)
diffic(x)
discrim(x)
```

`Jackknife sample mean` *Jackknife sample mean*

Description

Jackknife sample mean.

Usage

```
jack.mean(x)
```

Arguments

`x`	A numerical vector with data.

Details

An efficient implementation of the jackknife mean is provided.

Value

The jackknife sample mean.

Author(s)

Michail Tsagris

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

References

Efron Bradley and Robert J. Tibshirani (1993). An introduction to the bootstrap. New York: Chapman \& Hall/CRC.

See Also

`welch.tests`, `trim.mean`

Examples

```
x <- rnorm(50)
jack.mean(x)
```

`Kaplan-Meier estimate of a survival function`
Kaplan-Meier estimate of a survival function

Description

Kaplan-Meier estimate of a survival function.

Usage

```
km(ti, di)
```

Arguments

`ti`	A numerical vector with the survival times.
`di`	A numerical vector indicating the censorings. 0 = censored, 1 = not censored.

Details

The Kaplan-Meier estimate of the survival function takes place.

Value

A matrix with 4 columns. The non censored times, the number of subjects at risk, the number of events at each time and the estimated survival

Author(s)

Michail Tsagris

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

References

Kaplan E. L. and Meier P. (1958). Nonparametric estimation from incomplete observations. Journal of the American Statistical Association, 53(282): 457-481.

See Also

`sp.logiregs`

Examples

```
y <- rgamma(40, 10, 1)
di <- rbinom(40, 1, 0.6)
a <- km(y, di)
```

`Linear regression with clustered data` *Linear regression with clustered data*

Description

Linear regression with clustered data.

Usage

```
cluster.lm(y, x, id)
```

Arguments

`y`	The dependent variable, a numerical vector with numbers.
`x`	A matrix or a data.frame with the indendent variables.
`id`	A numerical variable with 1, 2, ... indicating the subject. Unbalanced design is of course welcome.

Details

A linear regression model for clustered data is fitted. For more information see Chapter 4.21 of Hansen (2019).

Value

A list including:

`be`	The (beta) regression coefficients.
`becov`	Robust covariance matrix of the regression coefficients.
`seb`	Robust standard errors of the regression coefficients.

Author(s)

Michail Tsagris

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

References

Hansen, B. E. (2019). Econometrics. https://www.ssc.wisc.edu/~bhansen/econometrics/Econometrics.pdf

See Also

`gee.reg`

Examples

```
y <- rnorm(200)
id <- sample(1:20, 200, replace = TRUE)
x <- rnorm(200, 3)
cluster.lm(y, x, id)
```

Mahalanobis depth *Mahalanobis depth*

Description

Mahalanobis depth.

Usage

```
depth.mahala(x, data)
```

Arguments

x	A numerical vector or matrix whose depth you want to compute.
data	A numerical matrix used to compute the depth of x.

Details

This function computes the Mahalanobis depth of x with respect to data.

Value

A numevrical vector with the Mahalanobis depth for each value of x.

Author(s)

Michail Tsagris

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

References

Mahalanobis P. (1936). On the generalized distance in statistics. Proceedings of the National Academy India, 12 49–55.

Liu R.Y. (1992). Data depth and multivariate rank tests. In Dodge Y. (editors), L1-Statistics and Related Methods, 279–294.

See Also

welch.tests, trim.mean

Examples

```
x <- as.matrix(iris[1:50, 1:4])
depth.mahala(x, x)
```

Many approximate simple logistic regressions

Many approximate simple logistic regressions.

Description

Many approximate simple logistic regressions.

Usage

```
sp.logiregs(y, x, logged = FALSE)
```

Arguments

`y`	The dependent variable, a numerical vector with 0s or 1s.
`x`	A matrix with the indendent variables.
`logged`	Should the p-values be returned (FALSE) or their logarithm (TRUE)?

Details

Many simple approximate logistic regressions are performed and hypothesis testing for the singificance of each coefficient is returned. The code is available in the paper by Sikorska et al. (2013). We simply took the code and made some minor modifications. The explanation and the motivation can be found in their paper. They call it semi-parallel logistic regressions, hence we named the function sp.logiregs.

Value

A two-column matrix with the test statistics (Wald statistic) and their associated p-values (or their loggarithm).

Author(s)

Initial author Karolina Sikorska. Modifications by Michail Tsagris.

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

References

Karolina Sikorska, Emmanuel Lesaffre, Patrick FJ Groenen and Paul HC Eilers (2013), 14:166. GWAS on your notebook: fast semi-parallel linear and logistic regression for genome-wide association studies.

https://bmcbioinformatics.biomedcentral.com/track/pdf/10.1186/1471-2105-14-166

See Also

logiquant.regs,bic.regs

Examples

```
y <- rbinom(200, 1, 0.5)
x <- matrix( rnorm(200 * 50), ncol = 50 )
a <- sp.logiregs(y, x)
```

Many Gamma regressions

Many Gamma regressions

Description

Many Gamma regressions.

Usage

```
gammaregs(y, x, tol = 1e-07, logged = FALSE, parallel = FALSE, maxiters = 100)
```

Arguments

y	The dependent variable, a numerical variable with non negative numbers for the Gamma and inverse Gaussian regressions. For the Gaussian with a log-link zero values are allowed.
x	A matrix with the indendent variables.
tol	The tolerance value to terminate the Newton-Raphson algorithm.
logged	A boolean variable; it will return the logarithm of the pvalue if set to TRUE.
parallel	Do you want this to be executed in parallel or not. The parallel takes place in C++, therefore you do not have the option to set the number of cores.
maxiters	The maximum number of iterations that can take place in each regression.

Details

Many simple Gamma regressions with a log-link are fitted.

Value

A matrix with the test statistic values and their relevant (logged) p-values.

Author(s)

Stefanos Fafalios and and Michail Tsagris

R implementation and documentation: Stefanos Fafalios <stefanosfafalios@gmail.com> and Michail Tsagris <mtsagris@uoc.gr>

References

McCullagh, Peter, and John A. Nelder. Generalized linear models. CRC press, USA, 2nd edition, 1989.

Zakariya Yahya Algamal and Intisar Ibrahim Allyas (2017). Prediction of blood lead level in maternal and fetal using generalized linear model. International Journal of Advanced Statistics and Probability, 5(2): 65-69.

See Also

bic.regs, gammareg

Examples

```
## Not run:
y <- rgamma(100, 3, 10)
x <- matrnorm(100, 10)
b <- glm(y ~ x[, 1], family = Gamma(log) )
anova(b, test= "F")
a <- gammaregs(y, x)
x <- NULL

## End(Not run)
```

`Many score based zero inflated Poisson regressions`
Many score based zero inflated Poisson regressions

Description

Many score based zero inflated Poisson regressions.

Usage

```
score.zipregs(y, x, logged = FALSE )
```

Arguments

`y`	A vector with discrete data, counts.
`x`	A matrix with data, the predictor variables.
`logged`	A boolean variable; it will return the logarithm of the pvalue if set to TRUE.

Details

Instead of maximising the log-likelihood via the Newton-Raphson algorithm in order to perform the hypothesis testing that $\beta_i = 0$ we use the score test. This is dramatcially faster as no model need to be fitted. The first derivative of the log-likelihood is known in closed form and under the null hypothesis the fitted values are all equal to the mean of the response variable y. The test is not the same as the likelihood ratio test. It is size correct nonetheless but it is a bit less efficient and less powerful. For big sample sizes though (5000 or more) the results are the same. It is also much faster then the classical likelihood ratio test.

Value

A matrix with two columns, the test statistic and its associated (logged) p-value.

Author(s)

Michail Tsagris

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

References

Lambert D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. Technometrics, 34(1):1-14.

Campbell, M.J. (2001). Statistics at Square Two: Understand Modern Statistical Applications in Medicine, pg. 112. London, BMJ Books.

See Also

ztp.reg,censpois.mle

Examples

```
x <- matrix( rnorm(1000 * 1000), ncol = 1000 )
y <- rpois(1000, 10)
y[1:150] <- 0
a <- score.zipregs(y, x)
x <- NULL
mean(a < 0.05) ## estimated type I error
```

Many simple quantile regressions using logistic regressions
Many simple quantile regressions using logistic regressions.

Description

Many simple quantile regressions using logistic regressions.

Usage

```
logiquant.regs(y, x, logged = FALSE)
```

Arguments

y	The dependent variable, a numerical vector.
x	A matrix with the indendent variables.
logged	Should the p-values be returned (FALSE) or their logarithm (TRUE)?

Details

Instead of fitting quantile regression models, one for each predictor variable and trying to assess its significance, Redden et al. (2004) proposed a simple singificance test based on logistic regression. Create an indicator variable I where 1 indicates a response value above its median and 0 elsewhere. Since I is binary, perform logistic regression for the predictor and assess its significance using the likelihood ratio test. We perform many logistic regression models since we have many predictors whose univariate association with the response variable we want to test.

Value

A two-column matrix with the test statistics (likelihood ratio test statistic) and their associated p-values (or their loggarithm).

Author(s)

Author: Michail Tsagris.

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

References

David T. Redden, Jose R. Fernandez and David B. Allison (2004). A simple significance test for quantile regression. Statistics in Medicine, 23(16): 2587-2597

See Also

bic.regs, sp.logiregs

Examples

```
y <- rcauchy(100, 3, 2)
x <- matrix( rnorm(100 * 50), ncol = 50 )
a <- logiquant.regs(y, x)
```

Many simple Weibull regressions
Many simple Weibull regressions.

Description

Many simple Weibull regressions.

Usage

```
weib.regs(y, x, tol = 1e-07, logged = FALSE, parallel = FALSE, maxiters = 100)
```

Arguments

y	The dependent variable, either a numerical variable with numbers greater than zero.
x	A matrix with the indendent variables.
tol	The tolerance value to terminate the Newton-Raphson algorithm.
logged	A boolean variable; it will return the logarithm of the pvalue if set to TRUE.
parallel	Do you want this to be executed in parallel or not. The parallel takes place in C++, and the number of threads is defined by each system's availiable cores.
maxiters	The maximum number of iterations that can take place in each regression.

Details

Many simple weibull regressions are fitted.

Value

A matrix with the test statistic values and their associated (logged) p-values.

Author(s)

Stefanos Fafalios

R implementation and documentation: Stefanos Fafalios <stefanosfafalios@gmail.com>.

See Also

bic.regs

Examples

```
y <- rgamma(100, 3, 4)
x <- matrix( rnorm( 100 * 30 ), ncol = 30 )
a <- weib.regs(y, x)
x <- NULL
```

`Many Welch tests` *Many Welch tests.*

Description

Many Welch tests.

Usage

```
welch.tests(y, x, logged = FALSE, parallel = FALSE)
```

Arguments

`y`	The dependent variable, a numerical vector.
`x`	A matrix with the indendent variables. They must be integer valued data starting from 1, not 0 and be consecutive numbers. Instead of a data.frame with factor variables, the user must use a matrix with integers.
`logged`	Should the p-values be returned (FALSE) or their logarithm (TRUE)?
`parallel`	If you want to run the function in parallel set this equal to TRUE.

Details

For each categorical predictor variable, a Welch test is performed. This is useful in feature selection algorithms, to determine for which variable, the means of the dependent variable differ across the different values.

Value

A two-column matrix with the test statistics (F test statistic) and their associated p-values (or their loggarithm).

Author(s)

Michail Tsagris

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

References

B.L. Welch (1951). On the comparison of several mean values: an alternative approach. Biometrika, 38(3/4), 330-336.

See Also

`sp.logiregs`, `pc.sel`

Examples

```
y <- rnorm(200)
x <- matrix(rbinom(200 * 50, 2, 0.5), ncol = 50) + 1
a <- welch.tests(y, x)
```

`Max-Min Parents and Children variable selection algorithm for continuous responses`
Max-Min Parents and Children variable selection algorithm for continuous responses

---

Description

Max-Min Parents and Children variable selection algorithm for continuous responses.

Usage

```
mmpc(y, x, max_k = 3, alpha = 0.05, method = "pearson",
ini = NULL, hash = FALSE, hashobject = NULL, backward = FALSE)
```

Arguments

`y`	The class variable. Provide a numeric vector.
`x`	The main dataset. Provide a numeric matrix.
`max_k`	The maximum conditioning set to use in the conditional independence test. Provide an integer. The default value set is 3.
`alpha`	Threshold for assessing p-values' significance. Provide a double value, between 0.0 and 1.0. The default value set is 0.05.
`method`	Currently only "pearson" is supported.
`ini`	This argument is used for the avoidance of the univariate associations re-calculations, in the case of them being present. Provide it in the form of a list.
`hash`	Boolean value for the activation of the statistics storage in a hash type object. The default value is false.
`hashobject`	This argument is used for the avoidance of the hash re-calculation, in the case of them being present, similarly to ini argument. Provide it in the form of a hash. Please note that the generated hash object should be used only when the same dataset is re-analyzed, possibly with different values of max_k and alpha.
`backward`	Boolean value for the activation of the backward/symmetry correction phase. This option removes and falsely included variables in the parents and children set of the target variable. It calls the `link{mmpc_bp}` for this purpose. The backward option seems dubious. Please do not use at the moment.

Details

The MMPC function implements the MMPC algorithm as presented in "Tsamardinos, Brown and Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm" http://www.dsl-lab.org/supplements/mmhc_paper/paper_online.pdf

Value

The output of the algorithm is an list including:

`selected`	The order of the selected variables according to the increasing pvalues.
`hashobject`	The hash object containing the statistics calculated in the current run.
`pvalues`	For each feature included in the dataset, this vector reports the strength of its association with the target in the context of all other variables. Particularly, this vector reports the max p-values found when the association of each variable with the target is tested against different conditional sets. Lower values indicate higher association.
`stats`	The statistics corresponding to the aforementioned pvalues (higher values indicate higher association).
`univ`	This is a list with the univariate associations; the test statistics and their corresponding logged p-values.
`max_k`	The max_k value used in the current execution.
`alpha`	The alpha value used in the current execution.
`n.tests`	If hash = TRUE, the number of tests performed will be returned. If hash != TRUE, the number of univariate associations will be returned.
`runtime`	The time (in seconds) that was needed for the execution of algorithm.

Author(s)

Marios Dimitriadis

R implementation and documentation: Marios Dimitriadis <kmdimitriadis@gmail.com>

References

Feature Selection with the R Package MXM: Discovering Statistically Equivalent Feature Subsets, Lagani, V. and Athineou, G. and Farcomeni, A. and Tsagris, M. and Tsamardinos, I. (2017). Journal of Statistical Software, 80(7).

Tsamardinos, I., Aliferis, C. F., & Statnikov, A. (2003). Time and sample efficient discovery of Markov blankets and direct causal relations. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 673-678). ACM.

Brown, L. E., Tsamardinos, I., & Aliferis, C. F. (2004). A novel algorithm for scalable and accurate Bayesian network learning. Medinfo, 711-715.

Tsamardinos, Brown and Aliferis (2006). The max-min hill-climbing Bayesian network structure learning algorithm. Machine learning, 65(1), 31-78.

Tsagis M. (2018). Guide on performing feature selection with the R package MXM. http://mensxmachina.org/site/wp-content/uploads/2018/04/Guide-on-performing-feature-selection-with-the-R-package-MXM.pdf

See Also

mmpc

Examples

```
set.seed(123)

# Dataset with continuous data
ds <- matrix(runif(100 * 500, 1, 100), ncol = 500)

# Class variable
tar <- 3 * ds[, 10] + 2 * ds[, 100] + 3 * ds[, 20] + rnorm(100, 0, 5)

mmpc(tar, ds, max_k = 3, alpha = 0.05, method = "pearson")
```

Max-Min Parents and Children variable selection algorithm for non continuous responses

Max-Min Parents and Children variable selection algorithm for non continuous responses

---

Description

Max-Min Parents and Children variable selection algorithm for non continuous responses.

Usage

```
mmpc2(y, x, max_k = 3, threshold = 0.05, test = "logistic", init = NULL,
tol = 1e-07, backward = FALSE, maxiters = 100, parallel = FALSE)
```

Arguments

y	The response variable, a numeric vector with either count data or binary data.
x	A numerical matrix with the independent (predictor) variables.
max_k	The maximum conditioning set to use in the conditional indepedence test (see Details). Integer, default value is 3.
threshold	Threshold (suitable values in (0, 1)) for assessing p-values significance. Default value is 0.05.
test	One of the following: "logistic", "poisson", "qpoisson".
init	A numeric vector with the logged p-values of the univariate associations. **Do not use this at the moment.**
tol	The tolerance value to stop the Newtn-Raphson algorithm inside a regression model.
backward	If TRUE, the backward (or symmetry correction) phase will be implemented. This removes any falsely included variables in the parents and children set of the target variable. It calls the `link{mmpcbackphase}` for this purpose.

`maxiters` The maximum number of iterations a Newtn-Raphson algorithm will perform inside a regression model.

`parallel` Do you want the computations to take part in parallel? Set TRUE if yes.

Details

MMPC tests each feature for inclusion (selection). It is a variant of the forward selection procedure. a) at every step it removes the non significant variables and does not check thema again. b) Instead of testing a candidate variable conditioning on all previously selected variables, it uses subsets of the previously selected variables. All possible subsets of maximum size equal to max_k. With the approrpiate pre-computations, at every step, it performs only the tests that were not exeucyted before, so it is not that time consuming.

Value

The output of the algorithm is an S3 object including:

`selectedVars` The selected variables, i.e., the signature of the target variable.

`pvalues` For each feature included in the dataset, this vector reports the strength of its association with the target in the context of all other variable. Particularly, this vector reports the max p-values found when the association of each variable with the target is tested against different conditional sets. Lower values indicate higher association.

`univ` A vector with the logged p-values of the univariate associations. This vector is very important for subsequent runs of MMPC with different hyper-parameters. After running SES with some hyper-parameters you might want to run MMPCagain with different hyper-parameters. To avoid calculating the univariate associations (first step) again, you can take this list from the first run of SES and plug it in the argument "ini" in the next run(s) of MMPC. This can speed up the second run (and subequent runs of course) by 50%. See the argument "univ" in the output values.

`kapa_pval` A list with the same number of elements as the max$_k$. Every element in the list is a matrix. The first column is the logged p-values, the second column is the variable whose conditional association with the response variable was tested and the other columns are the conditioning variables.

`max_k` The max_k option used in the current run.

`threshold` The threshold option used in the current run.

`n.tests` The number of tests performed by MMPC will be returned.

`runtime` The run time of the algorithm. A numeric vector. The first element is the user time, the second element is the system time and the third element is the elapsed time.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

References

Feature Selection with the R Package MXM: Discovering Statistically Equivalent Feature Subsets, Lagani, V. and Athineou, G. and Farcomeni, A. and Tsagris, M. and Tsamardinos, I. (2017). Journal of Statistical Software, 80(7).

Tsagis M. (2018). Guide on performing feature selection with the R package MXM. http://mensxmachina.org/site/wp-content/uploads/2018/04/Guide-on-performing-feature-selection-with-the-R-package-MXM.pdf

Tsamardinos, I., Aliferis, C. F., & Statnikov, A. (2003). Time and sample efficient discovery of Markov blankets and direct causal relations. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 673-678). ACM.

Brown, L. E., Tsamardinos, I., & Aliferis, C. F. (2004). A novel algorithm for scalable and accurate Bayesian network learning. Medinfo, 711-715.

See Also

mmpc, pc.sel, fbed.reg

Examples

```
y <- rbinom(100, 1, 0.5)
x <- matrix( rnorm(100 * 500), ncol = 500 )
m1 <- mmpc2(y, x, max_k = 3, threshold = 0.05, test = "logistic")
m2 <- fbed.reg(y, x, type = "logistic")
```

Maximum likelihood linear discriminant analysis

Maximum likelihood linear discriminant analysis

Description

Maximum likelihood linear discriminant analysis.

Usage

```
mle.lda(xnew, x, ina)
```

Arguments

xnew	A numerical vector or a matrix with the new observations, continuous data.
x	A matrix with numerical data.
ina	A numerical vector or factor with consecutive numbers indicating the group to which each observation belongs to.

Details

Maximum likelihood linear discriminant analysis is performed.

Value

A vector with the predicted group of each observation in "xnew".

Author(s)

Michail Tsagris

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

References

Kanti V. Mardia, John T. Kent and John M. Bibby (1979). Multivariate analysis. Academic Press, London.

See Also

welch.tests

Examples

```
x <- as.matrix(iris[, 1:4])
ina <- iris[, 5]
a <- mle.lda(x, x, ina)
```

Merge 2 sorted vectors in 1 sorted vector

Merge 2 sorted vectors in 1 sorted vector

Description

Merge 2 sorted vectors in 1 sorted vector.

Usage

```
Merge(x,y)
```

Arguments

x	A sorted vector with data.
y	A sorted vector with data.

Value

A sorted vector of the 2 arguments.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

See Also

is.lower.tri, is.upper.tri

Examples

```
x <- 1:10
y <- 1:20

Merge(x,y)

x <- y <- NULL
```

MLE of continuous univariate distributions defined on the positive line

MLE of continuous univariate distributions defined on the positive line

---

Description

MLE of continuous univariate distributions defined on the positive line.

Usage

```
halfcauchy.mle(x, tol = 1e-07)
powerlaw.mle(x)
```

Arguments

x	A vector with positive valued data (zeros are not allowed).
tol	The tolerance level up to which the maximisation stops; set to 1e-09 by default.

Details

Instead of maximising the log-likelihood via a numerical optimiser we have used a Newton-Raphson algorithm which is faster. See wikipedia for the equations to be solved. For the power law we assume that the minimum value of x is above zero in order to perform the maximum likelihood estimation in the usual way.

Value

Usually a list with three elements, but this is not for all cases.

iters	The number of iterations required for the Newton-Raphson to converge.
loglik	The value of the maximised log-likelihood.
scale	The scale parameter of the half Cauchy distribution.
alpha	The value of the power parameter for the power law distribution.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Manos Papadakis <papadakm95@gmail.com>.

References

N.L. Johnson, S. Kotz \& N. Balakrishnan (1994). Continuous Univariate Distributions, Volume 1 (2nd Edition).

N.L. Johnson, S. Kotz \& N. Balakrishnan (1970). Distributions in statistics: continuous univariate distributions, Volume 2

You can also check the relevant wikipedia pages for these distributions.

See Also

`zigamma.mle`,`censweibull.mle`

Examples

```
x <- abs( rcauchy(1000, 0, 2) )
halfcauchy.mle(x)
```

`MLE of distributions defined for proportions`
MLE of the Kumaraswamy distribution

Description

MLE of the Kumaraswamy distribution.

Usage

```
kumar.mle(x, tol = 1e-07, maxiters = 50)
simplex.mle(x, tol = 1e-07)
zil.mle(x)
```

Arguments

`x`	A vector with proportions or percentages. Zeros are allowed only for the zero inflated logistirc normal distribution (zil.mle).
`tol`	The tolerance level up to which the maximisation stops; set to 1e-07 by default.
`maxiters`	The maximum number of iterations the Newton-Raphson will perform.

Details

Instead of maximising the log-likelihood via a numerical optimiser we have used a Newton-Raphson algorithm which is faster. See wikipedia for the equations to be solved.

Value

Usually a list with three elements, but this is not for all cases.

`iters`	The number of iterations required for the Newton-Raphson to converge.
`param`	The two parameters (shape and scale) of the Kumaraswamy distribution or the means and sigma of the simpled distribution. For the zero inflated logistic normal, the probability of non zeros, the mean and the unbiased variance.
`loglik`	The value of the maximised log-likelihood.

Author(s)

Michail Tsagris

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

References

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. Journal of Hydrology. 46 (1-2): 79-88.

Jones, M.C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. Statistical Methodology. 6(1): 70-81.

Connie Stewart (2013). Zero-inflated beta distribution for modeling the proportions in quantitative fatty acid signature analysis. Journal of Applied Statistics, 40(5): 985-992.

Zhang, W. & Wei, H. (2008). Maximum likelihood estimation for simplex distribution nonlinear mixed models via the stochastic approximation algorithm. The Rocky Mountain Journal of Mathematics, 38(5): 1863-1875.

You can also check the relevant wikipedia pages.

See Also

`zigamma.mle`, `censweibull.mle`

Examples

```
u <- runif(1000)
a <- 0.4  ;  b <- 1
x <- ( 1 - (1 - u)^(1/b) )^(1/a)
kumar.mle(x)
```

MLE of some circular distributions with multiple samples
MLE of some circular distributions with multiple samples

Description

MLE of some circular distributions with multiple samples.

Usage

```
multivm.mle(x, ina, tol = 1e-07, ell = FALSE)
multispml.mle(x, ina, tol = 1e-07, ell = FALSE)
```

Arguments

x	A numerical vector with the circular data. They must be expressed in radians. For the "spml.mle" this can also be a matrix with two columns, the cosinus and the sinus of the circular data.
ina	A numerical vector with discrete numbers starting from 1, i.e. 1, 2, 3, 4,... or a factor variable. Each number denotes a sample or group. If you supply a continuous valued vector the function will obviously provide wrong results.
tol	The tolerance level to stop the iterative process of finding the MLEs.
ell	Do you want the log-likelihood returned? The default value is FALSE.

Details

The parameters of the von Mises and of the bivariate angular Gaussian distributions are estimated for multiple samples.

Value

A list including:

iters	The iterations required until convergence. This is returned in the wrapped Cauchy distribution only.
loglik	A vector with the value of the maximised log-likelihood for each sample.
mi	For the von Mises, this is a vector with the means of each sample. For the angular Gaussian (spml), a matrix with the mean vector of each sample
ki	A vector with the concentration parameter of the von Mises distribution at each sample.
gi	A vector with the norm of the mean vector of the angular Gaussian distribution at each sample.

Author(s)

Michail Tsagris

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

References

Mardia K. V. and Jupp P. E. (2000). Directional statistics. Chicester: John Wiley \& Sons.

Sra S. (2012). A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of Is(x). Computational Statistics, 27(1): 177-190.

Presnell Brett, Morrison Scott P. and Littell Ramon C. (1998). Projected multivariate linear models for directional data. Journal of the American Statistical Association, 93(443): 1068-1077.

Kent J. and Tyler D. (1988). Maximum likelihood estimation for the wrapped Cauchy distribution. Journal of Applied Statistics, 15(2): 247–254.

See Also

colspml.mle, purka.mle

Examples

```
y <- rcauchy(100, 3, 1)
x <- y
ina <- rep(1:2, 50)
multivm.mle(x, ina)
multispml.mle(x, ina)
```

MLE of some truncated distributions

MLE of some truncated distributions

Description

MLE of some truncated distributions.

Usage

```
trunccauchy.mle(x, a, b, tol = 1e-07)
truncexpmle(x, b, tol = 1e-07)
```

Arguments

x	A numerical vector with continuous data. For the Cauchy distribnution, they can be anywhere on the real line. For the exponential distribution they must be strictly positive.
a	The lower value at which the Cauchy distribution is truncated.
b	The upper value at which the Cauchy or the exponential distribution is truncated. For the exponential this must be greater than zero.
tol	The tolerance value to terminate the fitting algorithm.

Details

Maximum likelihood of some truncated distributions is performed.

Value

A list including:

iters	The number of iterations reuired by the Newton-Raphson algorithm.
loglik	The log-likelihood.
lambda	The λ parameter in the exponential distribution.
param	The location and scale parameters in the Cauchy distribution.

Author(s)

Michail Tsagris

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

References

David Olive (2018). Applied Robust Statistics (Chapter 4).

http://lagrange.math.siu.edu/Olive/ol-bookp.htm

See Also

purka.mle

Examples

```
x <- rnorm(500)
trunccauchy.mle(x, -1, 1)
```

MLE of the Cauchy distribution with zero location
MLE of the Cauchy distribution with zero location

Description

MLE of the Cauchy distribution with zero location

Usage

```
cauchy0.mle(x, tol = 1e-07)
```

Arguments

x	A numerical vector with positive real numbers.
tol	The tolerance level up to which the maximisation stops set to 1e-07 by default.

Details

The Cauchy is the t distribution with 1 degree of freedom. The cauchy0.mle estimates the usual Cauchy distribution, over the real line, but assumes a zero location.

Value

A list including:

iters	The number of iterations required by the Newton-Raphson algorithm.
loglik	The value of the maximissed log-likelihood.
scale	The estimated scale parameter.

Author(s)

Michail Tsagris

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

See Also

censweibull.mle

Examples

```
x <- abs( rcauchy(150, 0, 3) )
cauchy0.mle(x)
```

MLE of the censored Weibull distribution

MLE of the censored Weibull distribution

Description

MLE of the censored Weibull distribution.

Usage

```
censweibull.mle(x, di, tol = 1e-07)
```

Arguments

x	A vector with positive valued data (zeros are not allowed).
di	A vector of 0s (censored) and 1s (not censored) vales.
tol	The tolerance level up to which the maximisation stops; set to 1e-07 by default.

Details

Instead of maximising the log-likelihood via a numerical optimiser we have used a Newton-Raphson algorithm which is faster.

Value

A list including:

`iters`	The number of iterations required for the Newton-Raphson to converge.
`loglik`	The value of the maximised log-likelihood.
`param`	The vector of the parameters.

Author(s)

Michail Tsagris

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

References

Fritz Scholz (1996). Maximum Likelihood Estimation for Type I Censored Weibull Data Including Covariates. Technical report. ISSTECH-96-022, Boeing Information & Support Services, P.O. Box 24346, MS-7L-22.

http://faculty.washington.edu/fscholz/Reports/weibcensmle.pdf

See Also

km, censpois.mle

Examples

```
x <- rweibull(300, 3, 6)
censweibull.mle(x, di = rep(1, 300))
di <- rbinom(300, 1, 0.9)
censweibull.mle(x, di)
```

`MLE of the gamma-Poisson distribution`
MLE of the gamma-Poisson distribution

Description

MLE of the gamma-Poisson distribution.

Usage

```
gammapois.mle(x, tol = 1e-07)
```

Arguments

x	A numerical vector with positive data and zeros.
tol	The tolerance value to terminate the Newton-Raphson algorithm.

Details

MLE of the gamma-Poisson distribution is fitted. When the rate in the Poisson follows a gamma distribution with shape = r and scale θ, the resulting distribution is the gamm-Poisson. If the shape r is integer, the distribution is called negative binomial distribution.

Value

A list including:

iters	The iterations required by the Newton-Raphson to estimate the parameters of the distribution for the non zero data.
loglik	The full log-likelihood of the model.
param	The parameters of the model.

Author(s)

Michail Tsagris

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

References

Johnson Norman L., Kotz Samuel and Kemp Adrienne W. (1992). Univariate Discrete Distributions (2nd ed.). Wiley.

See Also

zigamma.mle

Examples

```
x <- rnbinom(200, 20, 0.7)
gammapois.mle(x)
```

MLE of the left censored Poisson distribution
MLE of the left censored Poisson distribution

Description

MLE of the left censored Poisson distribution.

Usage

```
censpois.mle(x, tol = 1e-07)
```

Arguments

`x`	A vector with positive valued data (zeros are not allowed).
`tol`	The tolerance level up to which the maximisation stops; set to 1e-07 by default.

Details

Instead of maximising the log-likelihood via a numerical optimiser we have used a Newton-Raphson algorithm which is faster. The lowest value in x is taken as the censored point. Values below are censored.

Value

A list including:

`iters`	The number of iterations required for the Newton-Raphson to converge.
`loglik`	The value of the maximised log-likelihood.
`lambda`	The estimated λ parameter.

Author(s)

Michail Tsagris

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

See Also

km, censweibull.mle

Examples

```
x1 <- rpois(10000,15)
x <- x1
x[x<=10] = 10
mean(x)
censpois.mle(x)$lambda
```

`MLE of the Purkayashta distribution` *MLE of the Purkayashta distribution*

Description

MLE of the Purkayashta distribution.

Usage

```
purka.mle(x, tol = 1e-07)
```

Arguments

`x`	A numerical vector with data expressed in radians or a matrix with spherical data.
`tol`	The tolerance value to terminate the Brent algorithm.

Details

MLE of the Purkayastha distribution is performed.

Value

A list including:

`theta`	The median direction.
`alpha`	The concentration parameter.
`loglik`	The log-likelihood.
`alpha.sd`	The standard error of the concentration parameter.

Author(s)

Michail Tsagris

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

References

Purkayastha S. (1991). A Rotationally Symmetric Directional Distribution: Obtained through Maximum Likelihood Characterization. The Indian Journal of Statistics, Series A, 53(1): 70-83

Cabrera J. and Watson G. S. (1990). On a spherical median related distribution. Communications in Statistics-Theory and Methods, 19(6): 1973-1986.

See Also

`circ.cor1`

Examples

```
x <- cbind( rnorm(100,1,1), rnorm(100, 2, 1) )
x <- x / sqrt(rowSums(x^2))
purka.mle(x)
```

MLE of the zero inflated Gamma and Weibull distributions
MLE of the zero inflated Gamma and Weibull distributions

Description

MLE of the zero inflated Gamma and Weibull distributions.

Usage

```
zigamma.mle(x, tol = 1e-07)
ziweibull.mle(x, tol = 1e-07)
```

Arguments

`x`	A numerical vector with positive data and zeros.
`tol`	The tolerance value to terminate the Newton-Raphson algorithm.

Details

MLE of some zero inflated models is performed.

Value

A list including:

`iters`	The iterations required by the Newton-Raphson to estimate the parameters of the distribution for the non zero data.
`loglik`	The full log-likelihood of the model.
`param`	The parameters of the model.

Author(s)

Michail Tsagris

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

References

Sandra Taylor and Katherine Pollard (2009). Hypothesis Tests for Point-Mass Mixture Data with Application to Omics Data with Many Zero Values. Statistical Applications in Geneticsand Molecular Biology, 8(1): 1–43.

Kalimuthu Krishnamoorthy, Meesook Lee and Wang Xiao (2015). Likelihood ratio tests for comparing several gamma distributions. Environmetrics, 26(8):571-583.

See Also

gammapois.mle

Examples

```
x <- rgamma(200, 4, 1)
x[sample(1:200, 20)] <- 0
zigamma.mle(x)
```

Moran's I measure of spatial autocorrelation
Moran's I measure of spatial autocorrelation

Description

Moran's I measure of spatial autocorrelation.

Usage

```
moranI(x, w, scaled = FALSE, R = 999)
```

Arguments

x	A numerical vector with observations.
w	The inverse of a (symmetric) distance matrix. After computing the distance matrix, you invert all its elements and the elements which are zero (diagonal) and have become Inf. set them to 0. This is the w matrix the functions requires. If you want an extra step, you can row standardise this matrix by dividing each row by its total. This will makw the rowsums equal to 1.
scaled	If the matrix is row-standardised (all rowsums are equal to 1) then this is TRUE and FALSE otherwise.
R	The number of permutations to use in order to obtain the permutation based-pvalue. If R is 1 or less no permutation p-value is returned.

Details

Moran' I index is a measure of spatial autocorrelation. that was proposed in 1950. Instead of computing an asymptotic p-value we compute a permutation based p-value utilizing the fast method of Chatzipantsiou et al. (2019).

Value

A vector of two values, the Moran's I index and its permutation based p-value. If R is 1 or less no permutation p-value is returned, and the second element is "NA".

Author(s)

Michail Tsagris

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

References

Moran, P. A. P. (1950). Notes on Continuous Stochastic Phenomena. Biometrika. 37(1): 17-23.

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 (To appear). https://arxiv.org/ftp/arxiv/papers/1806/1806.10947.pdf

See Also

censpois.mle, gammapois.mle

Examples

```
x <- rnorm(50)
w <- as.matrix( dist(iris[1:50, 1:4]) )
w <- 1/w
diag(w) <- 0
moranI(x, w, scaled = FALSE)
```

Multinomial regression

Multinomial regression

Description

Multinomial regression.

Usage

```
multinom.reg(y, x, tol = 1e-07, maxiters = 100)
```

Arguments

y	The response variable. A numerical or a factor type vector.
x	A matrix or a data.frame with the predictor variables.
tol	The tolerance value to terminate the Newton-Raphson algorithm.
maxiters	The maximum number of iterations Newton-Raphson will perform.

Value

A list including:

`iters`	The number of iterations required by the Newton-Raphson.
`loglik`	The value of the maximised log-likelihood.
`be`	A matrix with the estimated regression coefficients.

Author(s)

Michail Tsagris and Stefanos Fafalios

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Stefanos Fafalios <stefanosfafalios@gmail.com>.

References

Bohning, D. (1992). Multinomial logistic regression algorithm. Annals of the Institute of Statistical Mathematics, 44(1): 197-200.

See Also

`logiquant.regs`, `fbed.reg`

Examples

```
y <- iris[, 5]
x <- matrix( rnorm(150 * 3), ncol = 3 )
multinom.reg(y, x)
```

`Naive Bayes classifiers` *Naive Bayes classifiers*

Description

Naive Bayes classifiers.

Usage

```
weibull.nb(xnew = NULL, x, ina, tol = 1e-07)
normlog.nb(xnew = NULL, x, ina)
laplace.nb(xnew = NULL, x, ina)
```

Arguments

`xnew` A numerical matrix with new predictor variables whose group is to be predicted. For the Gaussian naive Bayes, this is set to NUUL, as you might want just the model and not to predict the membership of new observations. For the Gaussian case this contains any numbers, but for the multinomial and Poisson cases, the matrix must contain integer valued numbers only.

`x` A numerical matrix with the observed predictor variable values. For the Gaussian case this contains any numbers, but for the multinomial and Poisson cases, the matrix must contain integer valued numbers only.

`ina` A numerical vector with strictly positive numbers, i.e. 1,2,3 indicating the groups of the dataset. Alternatively this can be a factor variable.

`tol` The tolerance value to terminate the Newton-Raphson algorithm in the Weibull distribution.

Value

For the Weibull classifier a list including:

`shape` A matrix with the shape parameters.

`scale` A matrix with the scale parameters.

For the Gaussian with a log link (normlog) classifier a list including:

`expmu` A matrix with the mean parameters.

`sigma` A matrix with the (MLE, hence biased) variance parameters.

For the Laplace classifier a list including:

`location` A matrix with the location parameters (medians).

`scale` A matrix with the scale parameters.

`ni` The sample size of each group in the dataset.

`est` The estimated group of the xnew observations. It returns a numerical value back regardless of the target variable being numerical as well or factor. Hence, it is suggested that you do \"as.numeric(target)\" in order to see what is the predicted class of the new data.

Author(s)

Michail Tsagris

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

See Also

weibullnb.pred

Examples

```
x <- matrix( rweibull( 100, 3, 4 ), ncol = 2 )
ina <- rbinom(50, 1, 0.5) + 1
a <- weibull.nb(x, x, ina)
```

Naive Bayes classifiers for circular data

Naive Bayes classifiers for directional data

Description

Naive Bayes classifiers for directional data.

Usage

```
vm.nb(xnew = NULL, x, ina, tol = 1e-07)
spml.nb(xnew = NULL, x, ina, tol = 1e-07)
```

Arguments

`xnew`	A numerical matrix with new predictor variables whose group is to be predicted. Each column refers to an angular variable.
`x`	A numerical matrix with observed predictor variables. Each column refers to an angular variable.
`ina`	A numerical vector with strictly positive numbers, i.e. 1,2,3 indicating the groups of the dataset. Alternatively this can be a factor variable.
`tol`	The tolerance value to terminate the Newton-Raphson algorithm.

Details

Each column is supposed to contain angular measurements. Thus, for each column a von Mises distribution or an circular angular Gaussian distribution is fitted. The product of the densities is the joint multivariate distribution.

Value

A list including:

`mu`	A matrix with the mean vectors expressed in radians.
`mu1`	A matrix with the first set of mean vectors.
`mu2`	A matrix with the second set of mean vectors.
`kappa`	A matrix with the kappa parameters for the vonMises distribution or with the norm of the mean vectors for the circular angular Gaussian distribution.
`ni`	The sample size of each group in the dataset.
`est`	The estimated group of the xnew observations. It returns a numerical value back regardless of the target variable being numerical as well or factor. Hence, it is suggested that you do \"as.numeric(ina)\" in order to see what is the predicted class of the new data.

Author(s)

Michail Tsagris

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

See Also

`vmnb.pred`,`weibull.nb`

Examples

```
x <- matrix( runif( 100, 0, 1 ), ncol = 2 )
ina <- rbinom(50, 1, 0.5) + 1
a <- vm.nb(x, x, ina)
```

`Negative binomial regression`

Negative binomial regression

Description

Negative binomial regression.

Usage

```
negbin.reg(y, x, tol = 1e-07, maxiters = 100)
```

Arguments

`y`	The dependent variable, a numerical vector with integer valued numbers.
`x`	A matrix or a data.frame with the indendent variables.
`tol`	The tolerance value required by the Newton-Raphson to stop.
`maxiters`	The maximum iterations allowed.

Details

A negative binomial regression model is fitted. The standard errors of the regressions are not returned as we do not compute the full Hessian matrix at each step of the Newton-Raphson.

Value

A list including:

`be`	The regression coefficients.
`loglik`	The loglikelihood of the regression model.
`iters`	The iterations required by the Newton-Raphson.

Author(s)

Michail Tsagris

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Stefanos Fafalios <stefanosfafalios@gmail.com>.

See Also

ztp.reg

Examples

```
y <- rnbinom(100, 10, 0.7)
x <- matrix( rnorm(100 * 3), ncol = 3 )
mod <- negbin.reg(y, x)
```

Non linear least squares regression for percentages or proporions

Non linear least squares regression for percentages or proporions

---

Description

Non linear least squares regression for percentages or proporions.

Usage

```
propols.reg(y, x, cov = FALSE, tol = 1e-07 ,maxiters = 100)
```

Arguments

y	The dependent variable, a numerical vector with percentages or proporions, including 0s and or 1s.
x	A matrix with the indendent variables.
cov	Should the sandwich covariance matrix and the standard errors be returned? If yes, set this equal to TRUE.
tol	The tolerance value to terminate the Newton-Raphson algorithm. This is set to 10^{-7} by default.
maxiters	The maximum number of iterations that can take place during the fitting.

Details

The ordinary least squares between the observed and the fitted percentages is adopted as the objective function. This involves numerical optimization since the relationship is non-linear. There is no log-likelihood. This is the univariate version of the OLS regression for compositional data mentioned in Murteira and Ramalho (2016).

Value

A list including:

`sse`	The sum of squares of the raw residuals.
`be`	The beta coefficients.
`seb`	The standard errors of the beta coefficients, if the input argument argument was set to TRUE.
`covb`	The covariance matrix of the beta coefficients, if the input argument argument was set to TRUE.
`iters`	The number of iterations required by the Newton-Raphson algorithm.

Author(s)

Michail Tsagris

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

References

Murteira, Jose MR, and Joaquim JS Ramalho 2016. Regression analysis of multivariate fractional data. Econometric Reviews 35(4): 515-552.

See Also

`simplex.mle`,`kumar.mle`,`gee.reg`,`sp.logiregs`,`logiquant.regs`

Examples

```
y <- rbeta(150, 3, 4)
x <- iris
a <- propols.reg(y, x)
```

`Parametric bootstrap for linear regression model`
Parametric bootstrap for linear regression model

Description

Parametric bootstrap for linear regression model.

Usage

```
lm.parboot(x, y, R = 1000)
```

Arguments

x	The predictor variables, a vector or a matrix or a data frame.
y	The response variable, a numerical vector with data.
R	The number of parametric bootstrap replications to perform.

Details

An efficient implementation of the parametric bootstrap for linear models is provided.

Value

A matrix with R columns and rows equal to the number of the regression parameters. Each column contains the set of a bootstrap beta regression coefficients.

Author(s)

Michail Tsagris

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

References

Efron Bradley and Robert J. Tibshirani (1993). An introduction to the bootstrap. New York: Chapman \& Hall/CRC.

See Also

lm.drop1, leverage, pc.sel, mmpc

Examples

```
y <- rnorm(50)
x <- Rfast::matrnorm(50, 2)
a <- lm.parboot(x, y, 500)
```

Permutation t-test for 2 independent samples

Permutation t-test for 2 independent samples

Description

Permutation t-test for 2 independent samples.

Usage

```
perm.ttest2(x, y, B = 999)
```

Arguments

x	A numerical vector with the data.
y	A numerical vector with the data.
B	The number of permutations to perform.

Details

The usual permutation based p-value is computed.

Value

A vector with the test statistic and the permutation based p-value.

Author(s)

Michail Tsagris

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

References

Good P. I. (2005). Permutation, parametric and bootstrap tests of hypotheses: a practical guide to resampling methods for testing hypotheses. Springer 3rd Edition.

See Also

jack.mean,trim.mean,moranI

Examples

```
x <- rexp(30, 4)
y <- rbeta(30, 2.5, 7.5)
perm.ttest2(x, y, 999)
```

Prediction with some naive Bayes classifiers

Prediction with some naive Bayes classifiers

Description

Prediction with some naive Bayes classifiers.

Usage

```
weibullnb.pred(xnew, shape, scale, ni)
normlognb.pred(xnew, expmu, sigma, ni)
laplacenb.pred(xnew, location, scale, ni)
```

Arguments

xnew	A numerical matrix with new predictor variables whose group is to be predicted. For the Gaussian case this contain positive numbers only.
shape	A matrix with the group shape parameters. Each row corresponds to a group.
scale	A matrix with the group scale parameters. Each row corresponds to a group.
expmu	A matrix with the mean parameters.
sigma	A matrix with the (MLE, hence biased) variance parameters.
location	A matrix with the location parameters (medians).
ni	A vector with the frequencies of each group.

Value

A numerical vector with 1, 2, ... denoting the predicted group.

Author(s)

Michail Tsagris

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

See Also

weibull.nb

Examples

```
x <- matrix( rweibull( 100, 3, 4 ), ncol = 2 )
ina <- rbinom(50, 1, 0.5) + 1
a <- weibull.nb(x, x, ina)
est <- weibullnb.pred(x, a$shape, a$scale, a$ni)
table(ina, est)
```

Prediction with some naive Bayes classifiers for circular data

Prediction with some naive Bayes classifiers for circular data

---

Description

Prediction with some naive Bayes classifiers for circular data.

Usage

```
vmnb.pred(xnew, mu, kappa, ni)
spmlnb.pred(xnew, mu1, mu2, ni)
```

Arguments

xnew	A numerical matrix with new predictor variables whose group is to be predicted. Each column refers to an angular variable.
mu	A matrix with the mean vectors expressed in radians.
mu1	A matrix with the first set of mean vectors.
mu2	A matrix with the second set of mean vectors.
kappa	A matrix with the kappa parameters for the vonMises distribution or with the norm of the mean vectors for the circular angular Gaussian distribution.
ni	The sample size of each group in the dataset.

Details

Each column is supposed to contain angular measurements. Thus, for each column a von Mises distribution or an circular angular Gaussian distribution is fitted. The product of the densities is the joint multivariate distribution.

Value

A numerical vector with 1, 2, ... denoting the predicted group.

Author(s)

Michail Tsagris

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

See Also

vm.nb,weibullnb.pred

Examples

```
x <- matrix( runif( 100, 0, 1 ), ncol = 2 )
ina <- rbinom(50, 1, 0.5) + 1
a <- vm.nb(x, x, ina)
a2 <- vmnb.pred(x, a$mu, a$kappa, a$ni)
```

Principal component analysis

Principal component analysis

Description

Principal component analysis.

Usage

```
pca(x, center = TRUE, scale = TRUE, k = NULL, vectors = FALSE)
```

Arguments

x	A numerical $n \times p$ matrix with data where the rows are the observations and the columns are the variables.
center	Do you want your data centered? TRUE or FALSE.
scale	Do you want each of your variables scaled, i.e. to have unit variance? TRUE or FALSE.
k	If you want a specific number of eigenvalues and eigenvectors set it here, otherwise all eigenvalues (and eigenvectors if requested) will be returned.
vectors	Do you want the eigenvectors be returned? By dafault this is FALSE.

Details

The function is a faster version of R's prcomp.

Value

A list including:

values	The eigenvalues.
vectors	The eigenvectors.

Author(s)

Michail Tsagris

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

See Also

reg.mle.lda

Examples

```
x <- matrix( rnorm(1000 * 20 ), ncol = 20)
a <- pca(x)
x <- NULL
```

Random effects meta analysis

Random effects meta analysis

Description

Random effects meta analysis.

Usage

```
refmeta(yi, vi, tol = 1e-07)
```

Arguments

- `yi` The observations.
- `vi` This variances of the observations.
- `tol` The toleranve value to terminate Brent's algorithm.

Details

Random effects estimation, via restricted maximum likelihood estimation (REML), of the common mean.

Value

A vector with many elements. The fixed effects mean estimate, the $\bar{v}$ estimate, the I^2, the H^2, the Q test statistic and it's p-value, the τ^2 estimate and the random effects mean estimate.

Author(s)

Michail Tsagris

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

References

Annamaria Guolo1 and Cristiano Varin (2017). Random-effects meta-analysis: The number of studies matters. Statistical Methods in Medical Research, 26(3): 1500-1518. https://pdfs.semanticscholar.org/8df4/e5f0daf0c3e643fc228f680ded3cb35ddb9c.pdf

https://methods.cochrane.org/statistics/sites/methods.cochrane.org.statistics/files/public/uploads/SMG_training_course_2016/cochrane_smg_training_2016_viechtbauer.pdf

See Also

`bic.regs`

Examples

```
y <- rnorm(30)
vi <- rexp(30, 3)
refmeta(y, vi)
```

Regularised maximum likelihood linear discriminant analysis

Regularised maximum likelihood linear discriminant analysis

Description

Regularised maximum likelihood linear discriminant analysis.

Usage

```
reg.mle.lda(xnew, x, ina, lambda)
```

Arguments

`xnew`	A numerical vector or a matrix with the new observations, continuous data.
`x`	A matrix with numerical data.
`ina`	A numerical vector or factor with consecutive numbers indicating the group to which each observation belongs to.
`lambda`	A vector of regularization values λ such as (0, 0.1, 0.2,...).

Details

Regularised maximum likelihood linear discriminant analysis is performed. The function is not extremely fast, yet is pretty fast.

Value

A matrix with the predicted group of each observation in "xnew". Every column corresponds to a λ value. If you have just on value of λ, then you will have one column only.

Author(s)

Michail Tsagris

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

See Also

`mle.lda`, `welch.tests`

Examples

```
x <- as.matrix(iris[, 1:4])
ina <- iris[, 5]
a <- reg.mle.lda(x, x, ina, lambda = seq(0, 1, by = 0.1) )
```

Sample quantiles and col/row wise quantiles
Sample quantiles and col/row wise quantiles

Description

Sample quantiles and col/row wise quantiles.

Usage

```
colQuantile(x,probs,parallel=FALSE)
rowQuantile(x,probs,parallel=FALSE)
Quantile(x,probs)
```

Arguments

x	Numeric vector whose sample quantiles are wanted. NA and NaN values are not allowed in numeric vectors. For the col/row versions a numerical matrix.
probs	Numeric vector of probabilities with values in [0,1], not missing values. Values up to 2e-14 outside that range are accepted and moved to the nearby endpoint.
parallel	Do you want to do it in parallel in C++? TRUE or FALSE.

Details

This is the same function as R's built in "quantile" with its default option, **type = 7**. We have also implemented it in a col/row-wise fashion.

Value

The function will return a vector of the same mode as the arguments given. NAs will be removed.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

See Also

trim.mean

Examples

```
x<-rnorm(1000)
probs<-runif(10)
sum( quantile(x, probs = probs) - Quantile(x, probs) )
```

Score test for overdispersion in Poisson regression
Score test for overdispersion in Poisson regression

Description

Score test for overdispersion in Poisson regression.

Usage

```
overdispreg.test(y, x)
```

Arguments

y	A vector with count data.
x	A numerical matrix with predictor variables.

Details

A score test for overdispersion in Poisson regression is implemented.

Value

A vector with two values. The test statistic and its associated p-value.

Author(s)

Michail Tsagris

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

References

Yang Z., Hardin J.W. and Addy C.L. (2009). A score test for overdierpsdion in Poisson regression based on the generalised Poisson-2 model. Journal of Statistical Planning and Inference, 139(4): 1514–1521.

See Also

ztp.reg, censpois.mle wald.poisrat

Examples

```
y <- rnbinom(100, 10, 0.4)
x <- rnorm(100)
overdispreg.test(y, x)
```

`Single terms deletion hypothesis testin in a linear regression model`
Single terms deletion hypothesis testin in a linear regression model

---

Description

Single terms deletion hypothesis testin in a linear regression model.

Usage

```
lm.drop1(y, x, type = "F")
```

Arguments

`y`	The dependent variable, a numerical vector with numbers.
`x`	A numerical matrix with the indendent variables. We add, internally, the first column of ones.
`type`	If you want to perform the usual F (or t) test set this equal to "F". For the test based on the partial correlation set this equal to "cor".

Details

This is the same function as R's built in drop1 but it works with the F test and we have added a test based on the partial correlation coefficient. For the linear regression model, the Wald test is equivalent to the partial F test. So, instead of performing many regression models with single term deletions we perform one regression model with all variables and compute their Wald test effectively. Note, that this is true, only if the design matrix "x" contains the vectors of ones and in our case this must be, strictly, the first column. The second option is to compute the p-value of the partial correlation.

Value

A matrix with two columns. The test statistic and its associated pvalue for each independent variable.

Author(s)

Michail Tsagris

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

References

Hastie T., Tibshirani R. and Friedman J. (2008). The Elements of Statistical Learning (2nd Ed.), Springer.

See Also

`mmpc2`, `gee.reg`, `pc.sel`

Examples

```
y <- rnorm(150)
x <- as.matrix(iris[, 1:4])
a <- lm(y~., data.frame(x) )
drop1(a, test = "F")
lm.drop1(y, x )
```

Split the matrix in lower,upper triangular and diagonal
Split the matrix in lower,upper triangular and diagonal

Description

Split the matrix in lower,upper triangular and diagonal.

Usage

```
lud(x)
```

Arguments

x	A matrix with data.

Value

A list with 3 fields:

lower	The lower triangular of argument "x".
upper	The upper triangular of argument "x".
diagonal	The diagonal elements.

Author(s)

Manos Papadakis

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

See Also

Intersect

Examples

```
x <- matrix(runif(10*10),10,10)

b<-lud(x)
```

`Trimmed mean` *Trimmed mean*

Description

Trimmed mean.

Usage

```
trim.mean(x, a = 0.05)
colTrimMean(x, a = 0.05,parallel=FALSE)
rowTrimMean(x, a = 0.05,parallel=FALSE)
```

Arguments

`x`	A numerical vector or a numerical matrix.
`a`	A number in (0, 1), the proportion of data to trim.
`parallel`	Run the algorithm parallel in C++.

Details

The trimmed mean is computed. The lower and upper a% of the data are removed and the mean is calculated using the rest of the data.

Value

The trimmed mean.

Author(s)

Michail Tsagris and Manos Papadakis

R implementation and documentation: Michail Tsagris <mtsagris@yahoo.gr> and Manos Papadakis <papadakm95@gmail.com>

References

Wilcox R.R. (2005). Introduction to robust estimation and hypothesis testing. Academic Press.

See Also

`Quantile`

Examples

```
x <- rnorm(100, 1, 1)
all.equal(trim.mean(x, 0.05),mean(x, 0.05))

x<-matrix(x,10,10)

colTrimMean(x,0.05)
rowTrimMean(x,0.05)
```

`Variable selection using the PC-simple algorithm`
Variable selection using the PC-simple algorithm

Description

Variable selection using the PC-simple algorithm.

Usage

```
pc.sel(y, x, ystand = TRUE, xstand = TRUE, alpha = 0.05)
```

Arguments

`y`	A numerical vector with continuous data.
`x`	A matrix with numerical data; the independent variables, of which some will probably be selected.
`ystand`	If this is TRUE the response variable is centered. The mean is subtracted from every value.
`xstand`	If this is TRUE the independent variables are standardised.
`alpha`	The significance level.

Details

Variable selection for continuous data only is performed using the PC-simple algorithm (Buhlmann, Kalisch and Maathuis, 2010). The PC algorithm used to infer the skeleton of a Bayesian Network has been adopted in the context of variable selection. In other words, the PC algorithm is used for a single node.

Value

A list including:

`vars`	A vector with the selected variables.
`n.tests`	The number of tests performed.
`runtime`	The runtime of the algorithm.

Author(s)

Michail Tsagris

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

References

Buhlmann P., Kalisch M. and Maathuis M. H. (2010). Variable selection in high-dimensional linear models: partially faithful distributions and the PC-simple algorithm. Biometrika, 97(2): 261-278. https://arxiv.org/pdf/0906.3204.pdf

See Also

pc.skel, omp

Examples

```
y <- rnorm(100)
x <- matrix( rnorm(100 * 50), ncol = 50)
a <- pc.sel(y, x)
```

`Wald confidence interval for the ratio of two Poisson variables`
Wald confidence interval for the ratio of two Poisson variables

---

Description

Wald confidence interval for the ratio of two Poisson variables.

Usage

```
wald.poisrat(x, y, alpha = 0.05)
col.waldpoisrat(x, y, alpha = 0.05)
```

Arguments

x	A numeric vector or a matrix with count data.
y	A numeric vector or a matrix with count data.
alpha	The 1 - confidence level. The default value is 0.05.

Details

wald confidence interval for the ratio of two Poisson means is/are calculated.

Value

For the wald.poisrat a vector with three elements, the ratio and the lower and upper confidence interval limits. For the col.waldpoisrat a matrix with three columns, the ratio and the lower and upper confidence interval limits.

Author(s)

Michail Tsagris

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

References

Krishnamoorthy K., Peng J. and Zhang D. (2016). Modified large sample confidence intervals for Poisson distributions: Ratio, weighted average, and product of means. Communications in Statistics-Theory and Methods, 45(1): 83-97.

See Also

censpois.mle,

Examples

```
x <- rpois(100, 10)
y <- rpois(100, 10)
wald.poisrat(x, y)
```

Walter's confidence interval for the ratio of two binomial variables (and the relative risk)

Walter's confidence interval for the ratio of two binomial variables (and the relative risk)

Description

Walter's confidence interval for the ratio of two binomial variables (and the relative risk).

Usage

```
walter.ci(x1, x2, n1, n2, a = 0.05)
```

Arguments

x1	An integer number, greater than or equal to zero.
x2	A secondinteger number, greater than or equal to zero.
n1	An integer number, greater than or x1.
n2	A secondinteger number, greater than or equal to x2.
a	The significance level. The produced confidence interval has a confidence level equal to 1-a.

Details

This calculates a (1-a)% confidence interval for the ratio of two binomial variables (and hence for the relative risk) using Walter's suggestion (Walter, 1975). That is, to add 0.5 in each number. This not only overcomes the problem of zero values, but produces intervals that are more accurate than the classical asymptotic confidence interval (Alharbi and Tsagris, 2018).

Value

A list including:

`rat`	The ratio of the two binomial distributions.
`ci`	Walter's confidence interval.

Author(s)

Michail Tsagris

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

References

Walter S. (1975). The distribution of Levin's measure of attributable risk. Biometrika, 62(2): 371-372.

Alharbi N. and Tsagris M. (2018). Confidence Intervals for the Relative Risk. Biostatistics and Biometrics, 4(5). doi:10.19080/BBOAJ.2018.04.555647

https://juniperpublishers.com/bboaj/pdf/BBOAJ.MS.ID.555647.pdf

See Also

`mle.lda`, `welch.tests`

Examples

```
x1 <- rbinom(1, 20, 0.7)
x2 <- rbinom(1, 30, 0.6)
n1 <- 20
n2 <- 30
walter.ci(x1,x2,n1,n2)
```

`Zero truncated Poisson regression`

Zero truncated Poisson regression

Description

Zero truncated Poisson regression.

Usage

```
ztp.reg(y, x, full = FALSE, tol = 1e-07, maxiters = 100)
```

Arguments

y	The dependent variable, a numerical vector with integer valued numbers.
x	A matrix or a data.frame with the indendent variables.
full	If you want full information (standard errors, Walt test statistics and p-values of the regression coefficients) set this equal to TRUE.
tol	The tolerance value required by the Newton-Raphson to stop.
maxiters	The maximum iterations allowed.

Details

A zero truncated poisson regression model is fitted.

Value

A list including:

be	The regression coefficients if "full" was set to FALSE.
info	This is returned only if "full" was set to TRUE. It is a matrix with the regression coefficients, their standard errors, Walt test statistics and p-values.
loglik	The loglikelihood of the regression model.
iter	The iterations required by the Newton-Raphson.

Author(s)

Michail Tsagris

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

See Also

bic.regs

Examples

```
y <- rpois(100, 5)
y[y == 0] <- 1
x <- matrix( rnorm(100 * 5), ncol = 5 )
mod <- ztp.reg(y, x)
```

Index