

Package ‘Rfast2’

August 5, 2019

Type Package

Title A Collection of Efficient and Extremely Fast R Functions II

Version 0.0.4

Date 2019-08-05

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

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.

License GPL (>= 2.0)

LazyData TRUE

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-08-05 18:30:02 UTC

R topics documented:

Rfast2-package	3
Add many single terms to a model	4
Benchmark - Measure time	5
BIC of many simple univariate regressions	7
Check if a matrix is Lower or Upper triangular	8
Check whether a square matrix is skew-symmetric	9
Circular correlations between two circular variables	10
Column-wise means and variances	11
Column-wise MLE of some univariate distributions	12
Column-wise MLE of the angular Gaussian distribution	13

Column-wise pooled variances across groups	14
Column-wise summary statistics with grouping variables	15
Constrained least squares	16
Covariance between a variable and a matrix of variables	17
Diagonal values of the Hat matrix	18
Empirical entropy	19
Forward Backward Early Dropping selection regression	20
GEE Gaussian regression	21
Gumbel regression	23
Intersect	24
Item difficulty and discrimination	25
Kaplan-Meier estimate of a survival function	26
Linear regression with clustered data	27
Many approximate simple logistic regressions	28
Many simple quantile regressions using logistic regressions	29
Many simple Weibull regressions	30
Many Welch tests	31
Max-Min Parents and Children variable selection algorithm for continuous responses	32
Max-Min Parents and Children variable selection algorithm for non continuous responses	34
Maximum likelihood linear discriminant analysis	37
Merge 2 sorted vectors in 1 sorted vector	38
MLE of continuous univariate distributions defined on the positive line	39
MLE of distributions defined for proportions	40
MLE of some truncated distributions	41
MLE of the censored Weibull distribution	43
MLE of the gamma-Poisson distribution	44
MLE of the left censored Poisson distribution	45
MLE of the Purkayashta distribution	46
MLE of the zero inflated Gamma and Weibull distributions	47
Moran's I measure of spatial autocorrelation	48
Multinomial regression	49
Negative binomial regression	50
Principal component analysis	51
Random effects meta analysis	52
Regularised maximum likelihood linear discriminant analysis	54
Sample quantiles and col/row wise quantiles	55
Single terms deletion hypothesis testin in a linear regression model	56
Split the matrix in lower,upper triangular and diagonal	57
Trimmed mean	58
Variable selection using the PC-simple algorithm	59
Wald confidence interval for the ratio of two Poisson variables	60
Walter's confidence interval for the ratio of two binomial variables (and the relative risk)	61
Zero truncated Poisson regression	62

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: 0.0.4
Date: 2019-08-05
License: GPL-2

Maintainers

Manos Papadakis <papadakm95@gmail.com>

Note

Acknowledgments:

We would like to acknowledge Professor Kurt Hornik, Doctor Uwe Ligges (and the rest of R core team) for their invaluable help with this R package. Erich Studerus for his invaluable comments and Neal Fultz for his suggestions. Vassilis Vasdekis for his invaluable help with the random effects models. Marios Dimitriadis work was funded by the Special Account for Research Funds of the University of Crete, Department of Computer Science. Phillip Si is greatly acknowledged for his help with the Floyd-Warshal algorithm. Keefe Murphy for his invaluable help with NEWS file and for his suggestions. Zacharias Papadovassilakis gave us the inspiration for the memory efficient version of the k-NN algorithm. Yannis Pantazis explained us how the orhtogonal matching pursuit works. Achim Zeileis for his help with the quasi Poisson regression models. Pedro J. Aphalo for finding a bug. Dimitris Kyriakis for finding a bug.

Author(s)

Manos Papadakis <papadakm95@gmail.com>, Michail Tsagris <mtsagris@yahoo.gr>, Marios Dimitriadis <kmdimitriadis@gmail.com>, Stefanos Fafalios <stefanosfafalios@gmail.com>, Ioannis Tsamardinis <tsamard@csd.uoc.gr>, Matteo Fasiolo <matteo.fasiolo@gmail.com>, Giorgos Borboudakis <borbudak@gmail.com>, John Burkardt <jburkardt@fsu.edu> and Kleanthi Lakiotaki

<kliolak@gmail.com>, Changliang Zou <nk.chlzou@gmail.com>, Christina Chatzipantsiou <chatzipantsiou@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 independent 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")
```

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 expressions 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 independent variables.
family	The family of the regression models. "normal", "binomial", "poisson", "multinomial", "normlog" (Gaussian regression with log link), "spml" (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")
```

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 or 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
```

Circular correlations between two circular variables

Circular correlations between two circular variables

Description

Circular correlations between two circular variables.

Usage

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

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

Arguments

theta	The first circular variable expressed in radians, not degrees.
phi	The other circular 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-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

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

Details

This function calculates 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)
```

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 be percentages, excluding 0 and 1, whereas for the Borel distribution the `x` must contain integer values greater than 1.

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.

Author(s)

Michail Tsagris

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

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 be percentages, excluding 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 available 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 specifying whether you want the variances (FALSE) or the standard deviations (TRUE) of each column.

Details

This function calculates 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 specifying 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 minimize 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)
```

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)
```

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 with 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 significant 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.

Tsagris 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")
```

GEE Gaussian regression

GEE Gaussian regression

Description

GEE Gaussian regression.

Usage

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

Arguments

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

Details

Gaussian GEE regression is fitted.

Value

A list including:

<code>be</code>	The regression coefficients.
<code>seb</code>	The standard errors of the regression coefficients.
<code>phi</code>	The ϕ parameter.
<code>a</code>	The α parameter.
<code>covbeta</code>	The covariance matrix of the regression coefficients.
<code>iters</code>	The number of iterations 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](#), [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	<i>Gumbel regression</i>
-------------------	--------------------------

Description

Gumbel regression.

Usage

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

Arguments

<code>y</code>	The dependent variable, a numerical vector with real valued numbers.
<code>x</code>	A matrix or a data.frame with the independent variables.
<code>tol</code>	The tolerance value required by the Newton-Raphson to stop.
<code>maxiters</code>	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:

<code>be</code>	The regression coefficients.
<code>sigma</code>	The scale parameter.
<code>loglik</code>	The loglikelihood of the regression model.
<code>iters</code>	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)
```

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

<code>ti</code>	A numerical vector with the survival times.
<code>di</code>	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 independent 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)
```

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 independent 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 significance 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 logarithm).

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 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 independent 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 significance 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 logarithm).

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

<code>y</code>	The dependent variable, either a numerical variable with numbers greater than zero.
<code>x</code>	A matrix with the independent variables.
<code>tol</code>	The tolerance value to terminate the Newton-Raphson algorithm.
<code>logged</code>	A boolean variable; it will return the logarithm of the pvalue if set to TRUE.
<code>parallel</code>	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 available cores.
<code>maxiters</code>	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

<code>y</code>	The dependent variable, a numerical vector.
<code>x</code>	A matrix with the independent 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.
<code>logged</code>	Should the p-values be returned (FALSE) or their logarithm (TRUE)?
<code>parallel</code>	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 logarithm).

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 <code>link{mmhc_bp}</code> 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 independence 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 Newton-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 <code>link{mmpcbackphase}</code> for this purpose.
maxiters	The maximum number of iterations a Newton-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 them 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 appropriate pre-computations, at every step, it performs only the tests that were not executed 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 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.
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 MMPC again 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 subsequent runs of course) by 50%. See the argument "univ" in the output values.

<code>kapa_pval</code>	A list with the same number of elements as the <code>max_k</code> . 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.
<code>max_k</code>	The <code>max_k</code> option used in the current run.
<code>threshold</code>	The <code>threshold</code> option used in the current run.
<code>n_tests</code>	The number of tests performed by MMPC will be returned.
<code>runtime</code>	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).

Tsagris 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, maxiters = 50)
powerlaw.mle(x)
```

Arguments

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

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

<code>x</code>	A vector with proportions or percentages. Zeros are allowed only for the zero inflated logistic normal distribution (<code>zil.mle</code>).
<code>tol</code>	The tolerance level up to which the maximisation stops; set to 1e-07 by default.
<code>maxiters</code>	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.

<code>iters</code>	The number of iterations required for the Newton-Raphson to converge.
<code>param</code>	The two parameters (shape and scale) of the Kumaraswamy distribution or the means and sigma of the simplified distribution. For the zero inflated logistic normal, the probability of non zeros, the mean and the unbiased variance.
<code>loglik</code>	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 truncated distributions

MLE of some truncated distributions

Description

MLE of some truncated distributions.

Usage

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

Arguments

x	A numerical vector with continuous data. For the Cauchy distribution, 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 required 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)
truncauchy.mle(x, -1, 1)
```

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 gamma-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[zgamma.mle](#)**Examples**

```
x <- rbinom(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

```
zgamma.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 Genetics and 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
zgamma.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

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

Value

A list including:

<code>iters</code>	The number of iterations required by the Newton-Raphson.
<code>loglik</code>	The value of the maximised log-likelihood.
<code>be</code>	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](#), [fbcd.reg](#)

Examples

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

Negative binomial regression

Negative binomial regression

Description

Negative binomial regression.

Usage

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

Arguments

<code>y</code>	The dependent variable, a numerical vector with integer valued numbers.
<code>x</code>	A matrix or a data.frame with the independent variables.
<code>tol</code>	The tolerance value required by the Newton-Raphson to stop.
<code>maxiters</code>	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:

<code>be</code>	The regression coefficients.
<code>loglik</code>	The loglikelihood of the regression model.
<code>iters</code>	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)
```

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 default 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

<code>y</code>	The observations.
<code>vi</code>	This variances of the observations.
<code>tol</code>	The tolerance 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 its 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

<code>x</code>	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.
<code>probs</code>	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.
<code>parallel</code>	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) )
```

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	<i>Trimmed mean</i>
--------------	---------------------

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

Cox confidence interval for the ratio of two Poisson variables

Description

Cox 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

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

Details

A zero truncated poisson regression model is fitted.

Value

A list including:

<code>be</code>	The regression coefficients if "full" was set to FALSE.
<code>info</code>	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.
<code>loglik</code>	The loglikelihood of the regression model.
<code>iter</code>	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

- *Topic **Benchmark - Measure time**
 - Benchmark - Measure time, [5](#)
- *Topic **Check if a matrix is Lower or Upper triangular**
 - Check if a matrix is Lower or Upper triangular, [8](#)
- *Topic **Chrono Library**
 - Benchmark - Measure time, [5](#)
- *Topic **Column wise of grouping variables**
 - Column-wise summary statistics with grouping variables, [15](#)
- *Topic **Feature Selection**
 - Max-Min Parents and Children variable selection algorithm for continuous responses, [32](#)
- *Topic **Intersect**
 - Intersect, [24](#)
- *Topic **Merge 2 sorted vectors in 1 sorted vector**
 - Merge 2 sorted vectors in 1 sorted vector, [38](#)
- *Topic **Multinomial distribution**
 - Multinomial regression, [49](#)
- *Topic **Multiple Feature Signatures**
 - Max-Min Parents and Children variable selection algorithm for continuous responses, [32](#)
- *Topic **Sample Quantiles and col - row wise Quantiles**
 - Sample quantiles and col/row wise quantiles, [55](#)
- *Topic **Symmetric matrix**
 - Check whether a square matrix is skew-symmetric, [9](#)
- *Topic **Variable Selection**
 - Max-Min Parents and Children variable selection algorithm for continuous responses, [32](#)
- *Topic **regression**
 - Multinomial regression, [49](#)
- Add many single terms to a model, [4](#)
- add.term(Add many single terms to a model), [4](#)
- benchmark(Benchmark - Measure time), [5](#)
- Benchmark - Measure time, [5](#)
- BIC of many simple univariate regressions, [7](#)
- bic.regs, [5](#), [17](#), [18](#), [21](#), [29–31](#), [53](#), [63](#)
- bic.regs(BIC of many simple univariate regressions), [7](#)
- censpois.mle, [12](#), [43](#), [49](#), [61](#)
- censpois.mle(MLE of the left censored Poisson distribution), [45](#)
- censweibull.mle, [40](#), [41](#), [45](#)
- censweibull.mle(MLE of the censored Weibull distribution), [43](#)
- Check if a matrix is Lower or Upper triangular, [8](#)
- Check whether a square matrix is skew-symmetric, [9](#)
- cholesky, [9](#)
- circ.cor1, [47](#)
- circ.cor1(Circular correlations between two circular variables), [10](#)
- circ.cors1, [17](#)
- circ.cors1(Circular correlations between two circular variables), [10](#)
- Circular correlations between two circular variables, [10](#)
- cls(Constrained least squares), [16](#)
- cluster.lm, [22](#)
- cluster.lm(Linear regression with clustered data), [27](#)

- col.waldpoisrat (Wald confidence interval for the ratio of two Poisson variables), 60
- colborel.mle (Column-wise MLE of some univariate distributions), 12
- colGroup (Column-wise summary statistics with grouping variables), 15
- collogitnorm.mle (Column-wise MLE of some univariate distributions), 12
- collognorm.mle, 14
- collognorm.mle (Column-wise MLE of some univariate distributions), 12
- colmeansvars, 15, 25
- colmeansvars (Column-wise means and variances), 11
- colQuantile, 15
- colQuantile (Sample quantiles and col/row wise quantiles), 55
- colspml.mle (Column-wise MLE of the angular Gaussian distribution), 13
- colTrimMean (Trimmed mean), 58
- Column-wise means and variances, 11
- Column-wise MLE of some univariate distributions, 12
- Column-wise MLE of the angular Gaussian distribution, 13
- Column-wise pooled variances across groups, 14
- Column-wise summary statistics with grouping variables, 15
- Constrained least squares, 16
- cora, 9
- cova, 9
- covar, 22
- covar (Covariance between a variable and a matrix of variables), 17
- Covariance between a variable and a matrix of variables, 17
- Diagonal values of the Hat matrix, 18
- diffic (Item difficulty and discrimination), 25
- discrim (Item difficulty and discrimination), 25
- drop1, 56
- Empirical entropy, 19
- empirical.entropy (Empirical entropy), 19
- fbed.reg, 36, 50
- fbed.reg (Forward Backward Early Dropping selection regression), 20
- Forward Backward Early Dropping selection regression, 20
- gammapois.mle, 12, 14, 48, 49
- gammapois.mle (MLE of the gamma-Poisson distribution), 44
- GEE Gaussian regression, 21
- gee.reg, 17, 18, 21, 27, 56
- gee.reg (GEE Gaussian regression), 21
- Gumbel regression, 23
- gumbel.reg (Gumbel regression), 23
- halfcauchy.mle (MLE of continuous univariate distributions defined on the positive line), 39
- Intersect, 8, 24, 57
- intersect, 24
- is.lower.tri, 38
- is.lower.tri (Check if a matrix is Lower or Upper triangular), 8
- is.skew.symmetric (Check whether a square matrix is skew-symmetric), 9
- is.upper.tri, 38
- is.upper.tri (Check if a matrix is Lower or Upper triangular), 8
- Item difficulty and discrimination, 25
- Kaplan-Meier estimate of a survival function, 26
- km, 43, 45
- km (Kaplan-Meier estimate of a survival function), 26
- kumar.mle (MLE of distributions defined for proportions), 40
- leverage (Diagonal values of the Hat matrix), 18
- Linear regression with clustered data, 27

- lm.drop1 (Single terms deletion hypothesis test in a linear regression model), [56](#)
- logiquant.regs, [5](#), [21](#), [29](#), [50](#)
- logiquant.regs (Many simple quantile regressions using logistic regressions), [29](#)
- logistic_only, [7](#)
- lud (Split the matrix in lower, upper triangular and diagonal), [57](#)

- Many approximate simple logistic regressions, [28](#)
- Many simple quantile regressions using logistic regressions, [29](#)
- Many simple Weibull regressions, [30](#)
- Many Welch tests, [31](#)
- Max-Min Parents and Children variable selection algorithm for continuous responses, [32](#)
- Max-Min Parents and Children variable selection algorithm for non continuous responses, [34](#)
- Maximum likelihood linear discriminant analysis, [37](#)
- Merge (Merge 2 sorted vectors in 1 sorted vector), [38](#)
- Merge 2 sorted vectors in 1 sorted vector, [38](#)
- MLE of continuous univariate distributions defined on the positive line, [39](#)
- MLE of distributions defined for proportions, [40](#)
- MLE of some truncated distributions, [41](#)
- MLE of the censored Weibull distribution, [43](#)
- MLE of the gamma-Poisson distribution, [44](#)
- MLE of the left censored Poisson distribution, [45](#)
- MLE of the Purkayashta distribution, [46](#)
- MLE of the zero inflated Gamma and Weibull distributions, [47](#)
- mle.lda, [54](#), [62](#)
- mle.lda (Maximum likelihood linear discriminant analysis), [37](#)
- mmpc, [34](#), [36](#)
- mmpc (Max-Min Parents and Children variable selection algorithm for continuous responses), [32](#)
- mmpc2, [56](#)
- mmpc2 (Max-Min Parents and Children variable selection algorithm for non continuous responses), [34](#)
- Moran's I measure of spatial autocorrelation, [48](#)
- moranI (Moran's I measure of spatial autocorrelation), [48](#)
- multinom.reg (Multinomial regression), [49](#)
- Multinomial regression, [49](#)
- Negative binomial regression, [50](#)
- negbin.reg, [23](#)
- negbin.reg (Negative binomial regression), [50](#)

- omp, [60](#)
- pc.sel, [32](#), [36](#), [56](#)
- pc.sel (Variable selection using the PC-simple algorithm), [59](#)
- pc.skel, [60](#)
- pca (Principal component analysis), [51](#)
- poisson_only, [7](#)
- pooled.colVars, [11](#)
- pooled.colVars (Column-wise pooled variances across groups), [14](#)
- powerlaw.mle (MLE of continuous univariate distributions defined on the positive line), [39](#)
- pretty, [19](#)
- Principal component analysis, [51](#)
- print.benchmark (Benchmark - Measure time), [5](#)
- purka.mle, [42](#)
- purka.mle (MLE of the Purkayashta distribution), [46](#)

- Quantile, [6](#), [15](#), [19](#), [25](#), [58](#)
- Quantile (Sample quantiles and col/row wise quantiles), [55](#)

- Random effects meta analysis, [52](#)

- refmeta (Random effects meta analysis),
52
- reg.mle.lda, 52
- reg.mle.lda (Regularised maximum
likelihood linear
discriminant analysis), 54
- Regularised maximum likelihood linear
discriminant analysis, 54
- Rfast2-package, 3
- rowQuantile, 15
- rowQuantile (Sample quantiles and
col/row wise quantiles), 55
- rowTrimMean (Trimmed mean), 58
- Sample quantiles and col/row wise
quantiles, 55
- simplex.mle (MLE of distributions
defined for proportions), 40
- Single terms deletion hypothesis
testin in a linear regression
model, 56
- sp.logiregs, 5, 26, 30, 32
- sp.logiregs (Many approximate simple
logistic regressions), 28
- Split the matrix in lower, upper
triangular and diagonal, 57
- spml.reg, 10
- trim.mean, 6, 55
- trim.mean (Trimmed mean), 58
- Trimmed mean, 58
- truncauchy.mle (MLE of some truncated
distributions), 41
- truncexpmle (MLE of some truncated
distributions), 41
- Variable selection using the PC-simple
algorithm, 59
- Wald confidence interval for the ratio
of two Poisson variables, 60
- wald.poisrat (Wald confidence interval
for the ratio of two Poisson
variables), 60
- Walter's confidence interval for the
ratio of two binomial
variables (and the relative
risk), 61
- walter.ci (Walter's confidence
interval for the ratio of two
binomial variables (and the
relative risk)), 61
- weib.regs (Many simple Weibull
regressions), 30
- welch.tests, 22, 37, 54, 62
- welch.tests (Many Welch tests), 31
- Zero truncated Poisson regression, 62
- zgamma.mle, 40, 41, 45
- zgamma.mle (MLE of the zero inflated
Gamma and Weibull
distributions), 47
- zil.mle (MLE of distributions defined
for proportions), 40
- ziweibull.mle (MLE of the zero
inflated Gamma and Weibull
distributions), 47
- ztp.reg, 17, 18, 23, 51
- ztp.reg (Zero truncated Poisson
regression), 62