Package 'OOmisc'

February 19, 2015

Type Package

Title Ozgur-Ozlem Miscellaneous

Version 1.2

Date 2012-02-03

Author Ozgur Asar, Ozlem Ilk

Maintainer Ozgur Asar <o.asar@lancaster.ac.uk>

Description Includes miscellaneous functions.

License GPL (>= 2)

Repository CRAN

Date/Publication 2013-02-04 07:43:51

NeedsCompilation no

R topics documented:

OOmisc-package *Ozgur-Ozlem Miscellaneous*

Description

Includes miscellaneous functions. The package includes several functions with different goals such as random numer generation, numerical root finding, the calculation of accuracy measures and summary statistics etc.

Details

Package:	OOmisc
Version:	1.2
Date:	2012-02-03
License:	GPL (>=2)

accept.reject *Function to generate categorical random numbers from the specified psuedo-distribution by Accept-Reject Method.*

Description

Generates random numbers from the desired categorical distribution with given probabilities. Utilizes discrete uniform distribution.

Usage

```
accept.reject(prob, a, b, n)
```

Arguments

prob	a vector of probabilities
a	a numeric value which denotes the lower bound of the discrete uniform
b	a numeric value which denotes the upper bound of the discrete uniform
n	a numeric value for sample size

Details

b-a+1 should be equal to the number of levels of the desired variable.

Value

returns a vector of simulated sample of size n.

Author(s)

Ozgur Asar

Examples

```
accept.reject(prob=c(0.3,0.4,0.2,0.1),a=0,b=3,n=100)
```

`ci.prop` *A function to calculate exact and approximate confidence intervals for proportion.*

Description

Calculates exact and approximate confidence intervals for the proportion of a desired category level.

Usage

```
ci.prop(x, n, a)
```

Arguments

`x`	a numeric value which denotes the frequency of the desired category.
`n`	a numeric value which denotes the sample size.
`a`	a numeric value which denotes the significance level.

Value

Returns a matrix output which includes both the exact and approximate confidence intervals.

Author(s)

Ozlem Ilk, Ozgur Asar

References

Ilk, O. (2011). *R Yazilimina Giris* [Introduction to R Language]. ODTU Yayincilik [METU Press].

Examples

```
ci.prop(50,100,0.05)
```

`damped.newton` *A function to find the roots of univariate functions.*

Description

Finds roots of univariate functions by modifying the usual Newton-Raphson method by decreasing the step sizes when necessary.

Usage

```
damped.newton(fun, derf, x0, eps, maxit = 20, damp = seq(0, 40),silent=TRUE)
```

Arguments

`fun`	a function for which the root is searched.
`derf`	a function which is the first derivative of the function to be solved.
`x0`	a numeric value to be used to start the algorithm.
`eps`	a numeric value to be considered as the tolerance for convergence of the algorithm.
`maxit`	a numeric value which denotes maximum number of iterations to be consumed.
`damp`	a vector beginning from zero and increasing by one unit to decrease the step sizes.
`silent`	a logical statement which decides whether the iterations should be printed.

Value

Returns a numeric result of the root.

Author(s)

Ozgur Asar, Ozlem Ilk

References

Bose, K. S. (2008). *Numeric Computing in Fortran*. Alpha Science.

Conte, S. D., de Boor, C. (1980). *Elementary Numerical Analysis: An Algorithmic Approach, third edition*. New York: McGraw-Hill Publications.

Ilk, O. (2011). *R Yazilimina Giris* [Introduction to R Language]. ODTU Yayincilik [METU Press].

Examples

```
f1=function(x) x^3+sqrt(x)-1
df1=function(x) 3*x^2+(1/2)*x^(-1/2)
damped.newton(f1,df1,2,10^-10,maxit=40,silent=FALSE)
```

ePCP *Function to calculate expected proportion of correct prediction (ePCP).*

Description

Calculates ePCP and the related (1-α)% (approximate) confidence intervals for a given set of predicted success probabilities and the observed (binary) values.

Usage

```
ePCP(fit, y, alpha = 0.05)
```

Arguments

fit	a vector or matrix which includes the predicted success probabilities
y	a vector or matrix which includes the observed binary values
alpha	a numeric value for type I error

Value

Returns a matrix of output including the point estimate of the ePCP and the related (1-α)% confidence interval bounds

Author(s)

Ozgur Asar

References

Herron, M. (1999). Postestimation Uncertainty in Limited Dependent Variable Models. *Political Analysis*, **8**, 83–98.

Examples

```
fit<-runif(100)
y<-rbinom(100,1,0.5)
ePCP(fit,y,alpha=0.05)
```

expit *Function to do expit (exp(x)/(1+exp(x))) transformation*

Description

Calculates the expit transformation of a given set of values

Usage

```
expit(x)
```

Arguments

x	a vector or matrix which contains the values to be transformed

Details

expit is inverse of logit (log(x/(1-x)))

Value

Returns the transformed sample

Author(s)

Ozgur Asar

Examples

```
x<-rnorm(100)
expit(x)
```

fact *A function to calculate the factorial.*

Description

Calculates the factorial of a defined number.

Usage

```
fact(n)
```

Arguments

n	a numeric value for which the value of the factorial to be calculated.

Details

n should be an integer.

Value

Returns a numeric value.

Author(s)

Ozlem Ilk

References

Ilk, O. (2011). *R Yazilimina Giris* [Introduction to R Language]. ODTU Yayincilik [METU Press].

Examples

```
fact(4)
```

---

newton *A function to find the roots of univariate functions.*

Description

Finds roots of univariate functions by the usual Newton-Raphson (N-R) method.

Usage

```
newton(fun, derf, x0, eps, maxit = 20, silent = TRUE, tun=1)
```

Arguments

fun	a function for which the root is searched.
derf	a function which is the first derivative of the function to be solved.
x0	a numeric value to be used to start the algorithm.
eps	a numeric value to be considered as the tolerance for convergence of the algorithm.
maxit	a numeric value which denotes maximum number of iterations to be consumed.
silent	a logical statement which decides whether the iterations should be printed.
tun	a numeric value to decrease the steps

Details

tun is used to decrease the N-R steps, since it sometimes might miss the root value by taking large steps. tun=1 corresponds to usual N-R.

Value

Returns a numeric result of the root.

Author(s)

Ozlem Ilk, Ozgur Asar

References

Ilk, O. (2011). *R Yazilimina Giris* [Introduction to R Language]. ODTU Yayincilik [METU Press].

Examples

```
# function and the derivative
f1=function(x) x^3+sqrt(x)-1
df1=function(x) 3*x^2+(1/2)*x^(-1/2)
# searching for a reasonable initial
x0=seq(0,2,,100)
plot(x0,f1(x0),type="n")
lines(x0,f1(x0))
abline(h=0,lty=2)
newton(f1,df1,0.5,10**-10,silent=FALSE)
```

---

nll *Function to calculate negative log-likelihood (NLL)*

Description

Calculates NLL value of a given set of predicted success probabilities and observed (binary) values

Usage

```
nll(fit, y)
```

Arguments

fit	a vector or matrix which includes the predicted success probabilities
y	a vector or matrix which includes the observed binary values

Value

Returns a numeric value of the NLL estimate

Author(s)

Ozgur Asar

Examples

```
fit<-runif(100)
y<-rbinom(100,1,0.5)
nll(fit,y)
```

`rlaplace` *A function to generate random numbers from the Laplace distribution.*

Description

Generates a random sample of size n from the Laplace distribution with a desidered parameter, beta.

Usage

```
rlaplace(n, beta)
```

Arguments

`n`	a numeric value for sample size
`beta`	a numeric value for the parameter, beta

Details

beta should be positive.

Value

Returns a numeric vector of the sample.

Author(s)

Ozlem Ilk

References

Ilk, O. (2011). *R Yazilimina Giris* [Introduction to R Language]. ODTU Yayincilik [METU Press].

Examples

```
rlaplace(10,2)
```

`rtriangular` *A function to generate random numbers from the triangular distribution.*

Description

Generates a random sample of size n from the desired triangular distribution with parameters, a and b.

Usage

```
rtriangular(n, a, b)
```

Arguments

`n`	a numeric value which denotes the sample size.
`a`	a numeric value for the first parameter.
`b`	a numeric value for the second parameter.

Details

b should be greater than a.

Value

Returns a vector of sample.

Author(s)

Ozlem Ilk

References

Ilk, O. (2011). *R Yazilimina Giris* [Introduction to R Language]. ODTU Yayincilik [METU Press].

Examples

```
rtriangular(5,1,5)
```

secant *A function to find roots of univariate functions.*

Description

Finds the roots of univariate functions by using the Secant method.

Usage

```
secant(fun, x0, x1, eps, maxit = 20, silent = FALSE)
```

Arguments

fun	a function for which the root is searched.
x0	a numeric value to be used to start the algorithm (first initial).
x1	a numeric value to be used to start the algorithm (second initial).
eps	a numeric value to be considered as the tolerance for convergence of the algorithm.
maxit	a numeric value which denotes maximum number of iterations to be consumed.
silent	a logical statement which decides whether the iterations should be printed.

Value

Returns a numeric result.

Author(s)

Ozlem Ilk, Ozgur Asar

References

Ilk, O. (2011). *R Yazilimina Giris* [Introduction to R Language]. ODTU Yayincilik [METU Press].

Examples

```
## Example-1
f1=function(x) x^3+sqrt(x)-1
secant(f1,0.5,0.55,10^-10,silent=FALSE)

## Example-2
f2=function(x) x^3-sinh(x)+4*x^2+6*x+9
# searching for reasonable initials
x0=seq(-10,10,,100)
plot(x0,f2(x0),type="n")
lines(x0,f2(x0))
x0=seq(6,8,,100)
plot(x0,f2(x0),type="n")
```

```
lines(x0,f2(x0))
abline(h=0,lty=2)

secant(f2,7,7.2,10^-10,maxit=30,silent=FALSE)
```

---

sevennum *A function to calculate the 7-number summary.*

Description

Calculates 7-number summmary (minimum, 10th quantile, 25th quantile, median, 75th quantile, 90th quantile, maximum) of a data.

Usage

```
sevennum(x)
```

Arguments

x	a vector, matrix or data frame for the univariate dataset.

Value

Returns a matrix of the output.

Author(s)

Ozgur Asar

Examples

```
x<-rnorm(1000,0,1)
sevennum(x)
```

Index