Package 'bda'

January 5, 2018

Version 10.1.9

Date 2018-01-04

Title Density Estimation for Grouped Data

Author Bin Wang <bwang@southalabama.edu>.

Maintainer Bin Wang <bwang@southalabama.edu>

Depends R (>= 2.10.0)

Description Functions for density estimation based on grouped data, and next-generation gene expression data.

License Unlimited

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-01-05 04:56:38 UTC

R topics documented:

---

bda *Binned Data Analysis*

Description

This package collects algorithms for density estimation based on grouped (pre-binned) data.

---

bde *Density Estimation for Binned Data*

Description

Returns x and y coordinates of the density estimate of the probability density based on binned data.

Usage

```
bde(x, counts, nclass, breaks, bw,
    type="kde", from, to, gridsize=512L,
    lbound, conf.level)
```

Arguments

`x` A vector of sample data. 'NA' values will be automatically removed.

`counts` vector of frequencies (counts) of different bins. Missing values are not allowed.

`nclass` Number of classes

`breaks` vector of breaking points.

`bw` Bin (class) width.

`lbound` A numerical value showing where the distribution is bounded to the left. The distribution is not left bounded if `lbound` is missing.

`from,to,gridsize` parameters to define fine equally spaced grid points at which to estimate the density.

`type` Distribution family or smoothing type used to fit the histogram.

`conf.level` Confidence level for the pointwise/simultaneous confidence bands.

Details

Missing values are not allowed. A specific family of distribution is fitted to the a set of non-negative data that have binned. Families of distributions supported include:

`ewd`exponentiated Weibull distribution;

`weibull`Weibull distribution;

`dagum`Type I Dagum distribution.

Some histogram-based smoothing methods include:

`smkde,smoothkde`Blower and Kelsall's smooth KDE.

`histospline`Minnottee (1996,1998)'s histospline estimate.

`bootkde`Two-stage bootstrap KDE.

`npr/lpr/root-unroot`Estimate the density via local polynomial regression using root-unroot.

`fnmm/nmix/normmix/nm`Fitting finite normal mixture model using EM algorithm.

Value

a list containing the following components:

`x,y` vector of sorted `x` values at which the density estimate `y` was computed.

`type, npar, pars` If parametric method is used, return the type of distribution family in `type`, and estimated parameters in `pars`.

`ucb, lcb, conf.level` `conf.level` gives the confidence level; `lcb` and `ucb` are the corresponding confidence bands for the density function.

References

Wang, B. (2014). JSS paper.

Blower G, Kelsall J (2002). "Nonlinear kernel density estimation for binned data: convergence in entropy." Bernoulli, 8(4), 423-449.

Minnottee MC (1996). "The bias-optimized frequency polygon." Comput. Statist., 11, 35-48.

Minnottee MC (1998). "Achieving higher-order convergence rates for density estimation with binned data." JASA, 93(442), 663-672.

Wang, B. and Wertelecki, W. (2012) Density Estimation for Data With Rounding Errors. *Computational Statistics and Data Analysis*, doi: 10.1016/j.csda.2012.02.016.

Examples

```
data(hhi)
```

bin.NGS *Bin NGS data*

Description

To construct a histogram for NGS data.

Usage

```
bin.NGS(x, sLog,n.first=10)
```

Arguments

`x`	Next-Generation Sequencing data or similar. Could have a lot of zeros but negative values are not allowed
`sLog`	If `sLog` is missing, no sLog transformation. Otherwise, do `sLog(x)=log(x+c)`.
`n.first`	The number of first few NGS measurements for single-value classes.

Details

Histogram type density estimation via binning.

Author(s)

B. Wang <`bwang@southalabama.edu`>

References

AS 254, ...

Examples

```
# To be updated.
x = rexp(100,1)
x = c(rep(0,20),x)
out = fit.NGS(x)
plot(out)
```

---

binning *Data Binning*

Description

To bin a univariate data set in to a consecutive bins.

Usage

```
binning(x, counts, nclass, breaks, bw)
```

Arguments

x	A vector of raw data. 'NA' values will be automatically removed.
counts	Frequencies or counts of observations in different classes (bins)
nclass	Number of classes
breaks	The break points for data binning.
bw	Bin (class) width.

Details

Data binning can be done in the following cases: (1) both x and breaks are specified; (2) both counts and breaks are specified; (3) both x and bw are specified. If (3a) breaks is of length 1, it gives the lower limit of the first bin/class, and a sequence of equal-width bins will be defined. If (3b) breaks is missing, a random starting point will be chosen.

x and counts cannot be both given.

nclass will not be used if x is missing.

Examples

```
y <- c(10, 21, 56,79,114,122,110,85,85,61,47,49,47,44,31,20,11,4,4)
x <- 14.5 + c(0:length(y))
out1 <- binning(counts=y, breaks=x)
plot(out1)
```

compare *Compare two or more samples*

Description

An automated hypothesis testing procedure.

Usage

```
compare(y, x, var.name, alternative = c("two.sided", "less", "greater"),
        paired = FALSE, conf.level = 0.95, ...)
```

Arguments

`x,y` Two vector. 'NA' values will be automatically removed.

`var.name` Names of the two variables being compared for `x` and `y`, respectively.

`alternative,paired,conf.level,...` parameters for stat inferences.

Details

Will be updated later.

Author(s)

B. Wang <`bwang@southalabama.edu`>

cusum *Cumulative Sum Control Chart.*

Description

To construct the CUSUM plot.

Usage

```
cusum(x, ...)
ccusum(x,mu=NULL, k=0.5,h=4, ...)
bcusum(x,prob=NULL, R0=1.0,Ra=2, ...)
```

Arguments

`x` A sequnence of results.

`mu` mean values if 'x' is continous.

`h` A parameter to control recalculating the new mean to restart the CUSUM chart. Default: 4.

`k` Number of shift in sd unit, which controls the reference value K=k*s. Default: 1/2.

`prob` predicted probabilities for risk-adjusted CUSUM.

`R0, Ra` Odds ratios under H0 and Ha, respectively.Default: `R0=1`, `Ra=2`

`...` Controls

Details

To perform either continuous or binary CUSUM analysis based on the input "x".

In `bcusum`, the probability c0 under H0 is computed based on current data, which can also be given as a parameter `prob`.

Choose lower and upper limits h0 and h1: h1=1.5, 2.0, 2.5, 3, 10; h0=-2.5

Wald: h0 = -ln(1-alpha)/beta, h1 = ln(1-beta)/alpha. Alert lines: alpha=beta=0.1. Alarm/action lines: alpha=beta=0.01.

Rogers et al defined a scale adjustment ln(OR), say log(1.5)

de Leval et al: alert line with alpha=0.05, beta=0.2; alarm line with alpha=0.01, beta=0.1

Value

If the binary CUSUM is used, a class "BCUSUM" will be produced, otherwise, a class "CCUSUM" will be produced.

To detect increases we set `Ra > R0`, otherwise `Ra < R0`.

Note

Risk-adjusted CUSUM through the second parameter.

References

de Leval, M. R., Francois, K., Bull, C., Brawn, W., Spiegelhalter, D., Mar 1994. Analysis of a cluster of surgical failures. Application to a series of neonatal arterial switch operations. J. Thorac. Cardiovasc. Surg. 107, 914–923.

Page, E. S., 1954. Continuous inspection schemes. Biometrika 41 (1/2), pp. 100–115. URL http://www.jstor.org/stable/2333009

Parsonnet, V., Dean, D., Bernstein, A. D., Jun 1989. A method of uniform stratification of risk for evaluating the results of surgery in acquired adult heart disease. Circulation 79, 3–12.

Steiner, S. H., Cook, R. J., Farewell, V. T., Jan 1999. Monitoring paired binary surgical outcomes using cumulative sum charts. Stat Med 18, 69–86.

Steiner, S. H., Cook, R. J., Farewell, V. T., Treasure, T., Dec 2000. Monitoring surgical performance using risk-adjusted cumulative sum charts. Biostatistics 1, 441–452.

See Also

robot

Examples

```
data(robot)
hst = robot$TMHSTMIN
out = cusum(hst,mu=45)
plot(out)
```

Dagum *The Dagum Distribution*

Description

Density, distribution function, quantile function for the Dagum distribution.

Usage

```
ddagum(x, shape1,scale,shape2)
pdagum(q, shape1,scale,shape2)
qdagum(p, shape1,scale,shape2)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
shape1,scale,shape2	parameters.

Examples

```
ddagum(2, 1, 2, 1)
```

deg.NGS *A test of DEG detection*

Description

To test whether a gene differentially expressed in two experiments.

Usage

```
deg.NGS(x,y, gene,iter=1001)
```

Arguments

`x,y`	Two data files
`iter`	maximum iteration to find the optimal fit.
`gene`	name of gene to be tested

Details

We fit a mixture model to take care the zero values using component-0.

Author(s)

B. Wang <bwang@southalabama.edu>

References

AS 254, ...

Examples

```
# To be updated.
x = rexp(100,1)
x = c(rep(0,20),x)
out = fit.NGS(x)
plot(out)
```

eda — *Exploratory Data Analysis (EDA)*

Description

A graphical method for exploratory data analysis.

Usage

```
eda(x,plot=FALSE)
```

Arguments

x	A vector of numeric values. 'NA' values will be automatically removed.
plot	Show distribution and others using graph(s).

Details

This function is modified based on an S-Plus function written by Dr. Jiayang Sun (Course ST 426. Statistics Department, Case Western Reserve University). .

Author(s)

B. Wang <bwang@southalabama.edu>

References

J. W. Tukey. (1977). Exploratory Data Analysis. Addison-Wesley, New York.

Examples

```
x = rnorm(100)
eda(x)
```

ewd — *The Exponentiated Weibull Distribution*

Description

Density, distribution function, quantile function for the exponentiated Weibull distribution.

Usage

```
dewd(x, alpha, shape, scale)
pewd(q, alpha, shape, scale)
qewd(p, alpha, shape, scale)
```

Arguments

`x, q` vector of quantiles.

`p` vector of probabilities.

`alpha, shape, scale`
parameters.

Examples

```
dewd(2, 1, 2, 1)
```

`fit.GB` *Fitting GB*

Description

To fit a generalized beta distribution to raw/binned data.

Usage

```
fit.GB(x, lbound, ubound)
```

Arguments

`x` A vector of raw data, or a histogram or binned data.

`lbound,ubound` lower and upper bound for the support of the density. The bounds could be finite values, or positive or negative infinity.

Examples

```
x <- rnorm(100, 34.5,1.5)
hofc <- binning(x, bw=1)
lmd7 <- fit.GB(hofc)
plot(lmd7)
gof(lmd7)
```

`fit.GLD.FMKL` *Fitting FMKL GLD*

Description

To fit a FMKL GLD to raw/binned data.

Usage

```
fit.GLD.FMKL(x, lbound, ubound, percentile='exact', mle=FALSE)
fit.GLD(x, lbound, ubound, method='chisquare')
```

Arguments

`x`	A vector of raw data, or a histogram or binned data.
`percentile`	Use the exact percentiles (`exact`) or approxiated values (`approximate`).
`mle`	Logical. To find the MLE or not.
`lbound,ubound`	lower and upper bound for the support of the density. The bounds could be finite values, or positive or negative infinity.
`method`	Method for goodness-of-fit test.

Examples

```
data(hhi)
hmob <- binning(counts=hhi$mob, breaks=hhi$breaks)
lmd5 <- fit.GLD.FMKL(hmob)
lmd6 <-  fit.GLD.FMKL(hmob, mle=TRUE)
plot(lmd5)
lines(lmd6, col=4)
## GOP example (handbook) -- Hahn & Sapiro (1967)
## KS-GLD based on original data: (0.0345, 0.00009604, 0.87, 4.92)
## Table 3.6-1
breaks <- c(-Inf, seq(0.015, length=10, by=0.005), Inf)
counts <- c(1,9,30,44,58,45,29,17,9,4,4)
rho.mid <- c(0.0325, 0.0250, 0.667, 0.600)
rho.unif <- c(0.03352, 0.02531, 0.7786, 0.5009)
## histogram for chi-square test
## KS = 0.0225, p-value = 0.999.  Chi=0.5176, p-value=0.7720
breaks <- c(-Inf, 0.025, 0.03, 0.035, 0.04, 0.045, 0.05, Inf)
counts <- c(40,44,58,45, 29,17,17)
```

fit.lognormal *Fit a log-normal distribution to (binned) NGS data*

Description

Fit a log-normal distribution to (binned) NGS data

Usage

```
fit.lognormal(x, x.limits)
```

Arguments

x	Next-Generation Sequencing data or similar. Could have a lot of zeros but negative values are not allowed
x.limits	a low threshold of a non-zero measure which depends on a specific experiment.

Details

We fit a mixture model to take care the zero values using component-0.

Author(s)

B. Wang <bwang@southalabama.edu>

References

AS 254, ...

Examples

```
# To be updated.
x = rexp(100,1)
x = c(rep(0,20),x)
out = fit.NGS(x)
plot(out)
```

fit.mixnorm *Fit a Normal Mixture Model to binned data*

Description

Fit a finite normal mixture model for pre-binned data

Usage

```
fit.mixnorm(x,k,mu,s,p, x.range,lognormal=FALSE)
```

Arguments

`x` Ungrouped or grouped data

`k` number of components

`mu,s,p` Initial parameters of the Normal components.

`x.range` The range of `x`.

`lognormal` An indicator shows whether or not to fit a normal mixture for log-normal data.

Details

If the data are top-headed (or truncated), `nl, nu` will used to store the frequencies of the lower and uper bins. If the data are not truncated, the two variables will be used to return the predicted frequencies in these two bins instead.

Author(s)

B. Wang <bwang@southalabama.edu>

References

AS 254, ...

Examples

```
y <- c(10, 21, 56,79,114,122,110,85,85,61,47,49,47,44,31,20,11,4,4)
x <- 14.5 + c(0:length(y))
x.hist <- binning(counts=y, breaks=x)
mu <- c(20.0, 26.0)
p <- c(0.5, 0.5)
sig <- c(2.0, 2.8)
out <- fit.mixnorm(x.hist,mu=mu, p=p, s=sig)
```

fit.NGS *Fit a Normal Mixture Model to binned NGS data*

Description

Fit a finite normal mixture model for pre-binned NGS data

Usage

```
fit.NGS(x, iter.max=30,reflect=TRUE)
```

Arguments

`x` Next-Generation Sequencing data or similar. Could have a lot of zeros but negative values are not allowed

`iter.max` maximum iteration to find the optimal fit.

`reflect` Use reflect and replicate method to adjust for boundary effect.

Details

We fit a mixture model to take care the zero values using component-0.

Author(s)

B. Wang <bwang@southalabama.edu>

References

AS 254, ...

Examples

```
# To be updated.
x = rexp(100,1)
x = c(rep(0,20),x)
out = fit.NGS(x)
plot(out)
```

`fit.nmix.copula` *Fit a Bivariate Copula Based on Normal Mixture Model for NGS data*

Description

Fit bivariate copula for normal mixture models

Usage

```
fit.nmix.copula(x,y,mle.large=FALSE)
```

Arguments

`x,y`	Two Next-Generation Sequencing data with same structures
`mle.large`	Default ,`mle.large=FALSE`, don't use the big two-way histogram to find MLE.

Details

the two profiles need to be sent as parameters. We don't need to fit the normal mixture outside the function. But we need to fit normal mixture model for NGS data withinin the function to get the histograms, and the counts within two 2x2 tables. The output include 1) the two fitted model, and 2) estimated parameter Psi. In addition, we show 3) the Psi's used to search for the optimal Psi, and 4) Chi-square test statistics, and 5) the correponding p-values. In addition, we also find 6) the MLE of Psi, and 7) its Chi-square test statistic and 8) p-value.

Author(s)

B. Wang <bwang@southalabama.edu>

References

to be updated

Examples

```
# To be updated.
```

fit.Weibull *Two-parameter Weibull Distribution Maximum Likelihood Estimation*

Description

To compute the maximum likelihood estimates of the parameters of a 2-parameter Weibull distribution.

Usage

```
fit.Weibull(x, dist="Weibull")
```

Arguments

x	A vector of raw data, or a histogram or binned data.
dist	Distribution type: Weibull, GWD or EWD.

Examples

```
## raw data
x <- rweibull(100, 2, 1)
fit.Weibull(x)
##  binned data
data(hhi)
hmob <- binning(counts=hhi$mob, breaks=hhi$breaks)
fit.Weibull(hmob, dist="weibull")
fit.Weibull(hmob, dist="gwd")
fit.Weibull(hmob, dist="ewd")
```

GB *Generalized Beta Distribution*

Description

Density, distribution function, quantile function and random generation for the GB.

Usage

```
dgbeta(x, pars)
pgbeta(q, pars)
qgbeta(p, pars)
rgbeta(n, pars)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`pars`	Four parameters.

Value

`dgbeta` gives the density, `pgbeta` gives the distribution function, `qgbeta` gives the quantile function, and `rgbeta` generates random deviates.

Examples

```
dgbeta(.5, c(0,1,1,2))
```

`gld` *The mixed normal distribution*

Description

Density, distribution function, quantile function and random generation for the 4-parameter FMKL GLD with parameter `lambdas`.

Usage

```
dgld(x,lambdas)
pgld(q,lambdas)
qgld(p,lambdas)
rgld(n,lambdas)
```

Arguments

`x,q`	vector of quantiles.
`p`	A vector of probabilities.
`n`	number of observations. If 'length(n) > 1', the length is taken to be the number required.
`lambdas`	vector of dimension 4.

Author(s)

B. Wang `<bwang@southalabama.edu>`

Examples

```
lambdas <- c(33,1.4,-1.5,-.6)
rgld(3,lambdas)
qgld(c(0,.1,.3,.5,.8,.9,1),lambdas)
pgld(c(25,30,34,35,40), lambdas)
dgld(c(25,30,34,35,40), lambdas)
```

gof *Goodness of fit test*

Description

To perform goodness of fit test.

Usage

```
gof(x, dist, pars, method='ks')
```

Arguments

`x`	sample data.
`dist,pars`	Target population distribution type and parameters.
`method`	Two options: KS-test or Chi.square test.

Examples

```
data(hhi)
hmob <- binning(counts=hhi$mob, breaks=hhi$breaks)
lmd5 <- fit.GLD.FMKL(hmob)
gof(lmd5)
```

gwd *The Exponentiated Weibull Distribution*

Description

Density, distribution function, quantile function for the exponentiated Weibull distribution.

Usage

```
dgwd(x, alpha, sigma, lambda)
pgwd(q, alpha, sigma, lambda)
qgwd(p, alpha, sigma, lambda)
rgwd(n, alpha, sigma, lambda)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	sample size
`alpha, sigma, lambda`	parameters.

Examples

```
  dewd(2, 1, 2, 1)
```

`hhi` *Mobile 2010 Household income*

Description

Prebinned data for 2010 Mobile household income.

Usage

```
data(hhi)
```

References

URL: http://www.clrsearch.com/Mobile-Demographics/AL/Household-Income

Examples

```
data(hhi)
hhi
```

`laplace` *The Laplace (Double Exponential) Distribution*

Description

Density, distribution function, quantile function and random generation for the Laplace (double exponential) distribution with mean `mu` and rate `rate` (i.e., mean `1/rate`).

Usage

```
dlap(x, mu=0, rate=1)
plap(q, mu=0, rate=1)
qlap(p, mu=0, rate=1)
rlap(n, mu=0, rate=1)
fitlap(x,mu)
```

Arguments

`x, q` vector of quantiles.

`p` vector of probabilities.

`n` number of observations. If `length(n) > 1`, the length is taken to be the number required.

`mu` vector of means.

`rate` vector of rates.

Details

If `mu` is not specified, it assumes the default value of `0`.

If `rate` is not specified, it assumes the default value of `1`.

Value

`dlap` gives the density, `plap` gives the distribution function, `qlap` gives the quantile function, and `rlap` generates random deviates.

See Also

`exp` for the exponential function.

Distributions for other standard distributions, including `dgamma` for the gamma distribution and `dexp` for the exponential distribution. `dweibull` for the Weibull distribution, both of which generalize the exponential.

Examples

```
dlap(100)
x <- rlap(1000,mu=0, rate=0.5)
x0 = seq(-7,7, length=100)
fitlap(x)
plot(density(x), xlim=c(-7,7), ylim=c(0,0.35))
(out = fitlap(x))
lines(dlap(x0, out$mu, out$rate)~x0, col=2, lty=2)
(out = fitlap(x, mu=0))
lines(dlap(x0, out$mu, out$rate)~x0, col=4, lty=2)
```

ld50.logit *Predict Doses for Binomial Assay model (using counts)*

Description

Calibrate binomial assays, generalizing the calculation of LD50 based on a logistic regression model.

Usage

```
ld50.logit(ndead, ntotal, dose, cf = 1:2, p = 0.5)
```

Arguments

ndead	A vector of number of failures.
ntotal	Total number of trials.
dose	A vector of dosages.
cf	The terms in the coefficient vector giving the intercept and coefficient of (log-)dose
p	Probabilities at which to predict the dose needed.

References

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* Springer.

Examples

```
ldose <- rep(0:5, 2)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
n=20

ld50.logit(numdead,n,ldose,p = 0.5)
```

ld50.logitfit *Predict Doses for Binomial Assay model (using counts)*

Description

Calibrate binomial assays, generalizing the calculation of LD50 based on a logistic regression model.

Usage

```
ld50.logitfit(rate, dose, p = 0.5)
```

Arguments

`rate` A vector of percentages of successes among all trials.

`dose` A vector of dosages.

`p` Probabilities at which to predict the dose needed.

References

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* Springer.

Examples

```
ldose <- rep(0:5, 2)
rate <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)/20

ld50.logitfit(rate,ldose,p = 0.5)
```

`lps.variance` *compute the variance of the local polynomial regression function*

Description

To compute the variance of the local polynomial regression function

Usage

```
lps.variance(y,x,gridsize, bw, method="Rice")
```

Arguments

`y,x` Two numerical vectors: `y` is the response and `x` is the predictor.

`bw` Smoothing parameter. Is used only when `method='Wasserman'` or `method='heteroscedastic'`.

`gridsize` The size of a vector of grid points where the variance of `r(x)` is to be estimated. This parameter is needed only when `method='heteroscedastic'`.

`method` We use four method to compute the variance of r(x): Method 1) Larry Wasserman– nearly unbiased. This method based on an lps object; Method 2) Rice 1984 Method 3) Gasser et al (1986) – a variation of method 3. Method 4) For heteroscedastic errors. Need to estimate based on an lpr object. Yu and Jones (2004). Defaulty method: `Rice`.

Author(s)

B. Wang <`bwang@southalabama.edu`>

Examples

```
n = 100
x=rnorm(n)
y=x^2+rnorm(n)
bw = lps.variance
par(mfrow=c(1,1))
out=lpsmooth(y,x)
#plot(out, scb=TRUE, type='l')
vrx = lps.variance(y,x)
out=lpsmooth(y,x,sd.y=sqrt(vrx), bw=0.5)
plot(y~x, pch='.')
lines(out, scb=TRUE, col=2)

x0 = seq(min(x),  max(x), length=100)
y0 = x0^2
lines(y0~x0, col=4)
```

`lpsmooth` *non-parametric regression*

Description

To fit nonparametric regression model.

Usage

```
 lpsmooth(y,x, bw, sd.y,lscv=FALSE, adaptive=FALSE,
    from, to, gridsize,conf.level=0.95)
```

Arguments

`y,x` Two numerical vectors.

`from,to,gridsize` start point, end point and size of a fine grid where the EDF will be evaluated.

`bw` Smoothing parameter. Numeric or character value is allowed. If missing, adaptive (LSCV) bandwidth selector will be used.

`lscv,adaptive` If `lscv = FALSE`, use the given bandwidth to fit lpr directly. If `lscv = TRUE` and `adaptive = FALSE`, compute lscv bandwidth and fit lpr. Initial bandwidth should be given. If `lscv = TRUE` and `adaptive = TURE`, compute lscv bandwidth, then compute varying smoothing parameter, then fit lpr. This algorithm could be extremeely slow when the sample size is very large.

`sd.y` Standard deviation of `y`.

`conf.level` Confidence level.

Author(s)

B. Wang <bwang@southalabama.edu>

Examples

```
x <- rnorm(100,34.5,1.5)
e <- rnorm(100,0,2)
y <- (x-32)^2 + e
out <- lpsmooth(y,x)
out
plot(out, type='l', scb=TRUE)
x0 <- seq(min(x),max(x),length=100)
y0 <- (x0-32)^2
lines(x0, y0, col=2)
points(x, y, pch="*", col=4)
```

---

mediation.test *The Sobel mediation test*

Description

To compute statistics and p-values for the Sobel test. Results for three versions of "Sobel test" are provided: Sobel test, Aroian test and Goodman test.

Usage

```
mediation.test(mv,iv,dv)
```

Arguments

mv	The mediator variable.
iv	The independent variable.
dv	The dependent variable.

Details

To test whether a mediator carries the influence on an IV to a DV.

Value

Missing values are not allowed.

Author(s)

B. Wang <bwang@southalabama.edu>

References

MacKinnon, D. P., & Dwyer, J. H. (1993). Estimating mediated effects in prevention studies. *Evaluation Review*, 17, 144-158.

MacKinnon, D. P., Warsi, G., & Dwyer, J. H. (1995). A simulation study of mediated effect measures. *Multivariate Behavioral Research*, 30, 41-62.

Preacher, K. J., & Hayes, A. F. (2004). SPSS and SAS procedures for estimating indirect effects in simple mediation models. *Behavior Research Methods,Instruments, & Computers*, 36, 717-731.

Preacher, K. J., & Hayes, A. F. (2008). asymptotic and resampling strategies for assessing and comparing indirect effects in multiple mediator models. *Behavior Research Methods, Instruments, & Computers*, 40, 879-891.

Examples

```
mv = rnorm(100)
iv = rnorm(100)
dv = rnorm(100)
mediation.test(mv,iv,dv)
```

metamean.test *Test Equal Means*

Description

To test the equality of two means based on two sequences of sample means.

Usage

```
metamean.test(x,n,group,s,mu, sigma,years,
  alternative = c("two.sided", "less", "greater"),
  conf.level=0.95, sizable=FALSE,legend.pos=2)
```

Arguments

x,n,s	Vectors of sample means, sample sizes, and sample standard deviations.s can be missing.
mu, sigma	The vectors of the mean and standard deviation values for the populations or much larger sample.
years	The years of the collected data. If missing, replace with a sequence of integers starting with 1.
group	A vector showing the group IDs. Must have two levels.
alternative	Type of test.
conf.level	Confidence level if the means.
sizable	A logical variable to control the size of the point symbols.
legend.pos	Position of the legend: 1=top right, 2=top left, 3=bottom left, 4=bottom right, 0=no legend.

Details

To be updated.

Author(s)

B. Wang <bwang@southalabama.edu>

mixnorm *The mixed normal distribution*

Description

Density, distribution function, quantile function and random generation for the normal mixture distribution with means equal to 'mu' and standard deviations equal to 's'.

Usage

```
dmixnorm(x,p,mean,sd)
pmixnorm(q,p,mean,sd)
qmixnorm(prob,p,mean,sd)
rmixnorm(n,p,mean,sd)
```

Arguments

`x,q`	vector of quantiles in dmixnorm and pmixnorm. In qmixnorm, 'x' is a vector of probabilities.
`p`	proportions of the mixture components.
`prob`	A vector of probabilities.
`n`	number of observations. If 'length(n) > 1', the length is taken to be the number required.
`mean`	vector of means
`sd`	vector of standard deviations

Author(s)

B. Wang <bwang@southalabama.edu>

Examples

```
p <- c(.4,.6)
mu <- c(1,4)
s <- c(2,3)
dmixnorm(c(0,1,2,20),p,mu,s)
pmixnorm(c(0,1,2,20),p,mu,s)
qmixnorm(c(0,1,.2,.20),p,mu,s)
rmixnorm(3,p,mu,s)
```

normalize.NGS *To normalize a NGS profile using the mixture model method*

Description

To normalize a NGS profile using the mixture model method

Usage

```
normalize.NGS(x,y,method='mixture')
```

Arguments

`x,y` Two gene profiles to be normalized

`method` choose normalization method. Default: `mixture`, by fitting finite normal mixture models; `qunatile`, quantile normalization.

Author(s)

B. Wang <bwang@southalabama.edu>

References

TBA ...

Examples

```
# To be updated.
```

oddsratio *Odds Ratio and Relative Risk*

Description

To compute the odds ratio and relative risk based on a 2 X 2 table.

Usage

```
oddsratio(x,alpha=0.05,n,...)
```

Arguments

`x` A vector of length 2 of the number of events from the case and control studies.

`n` A vector of length 2 of the sample sizes.

`alpha` The significance level. Default: 0.05.

`...` Controls

Details

`x` can be a matrix or a data.frame: the first columns showing the number of events and the second column showing the sample sizes.

Exact confidence limits for the odds ratio by using an algorithm based on Thomas (1971). See also Gart (1971). If the sample sizes are too large, the exact confidence interval may not work due to overflow problem.

Asymptotic confidence limits are computed according to SAS/STAT(R) 9.2 User's Guide, Second Edition.

Score method: code has been published for generating confidence intervals by inverting a score test. It is available from http://web.stat.ufl.edu/~aa/cda/R/two_sample/R2/

See also "riskratio" and "oddsratio" in R package *epitools*.

Value

`OR`	an estimate of odds ratio;
`RR`	an estimate of realtive risk;
`ORCI`	A table showing various (1-alpha)% confidence limits for OR;
`RRCI`	A table showing various (1-alpha)% confidence limits for RR;

References

Agresti, A. (1990) _Categorical data analysis_. New York: Wiley. Pages 59-66.

Agresti, A. (1992), A Survey of Exact Inference for Contingency Tables Statistical Science, Vol. 7, No. 1. (Feb., 1992), pp. 131-153.

Agresti, A. (2002), Categorical Data Analysis, Second Edition, New York: John Wiley \& Sons.

Fisher, R. A. (1935) The logic of inductive inference. _Journal of the Royal Statistical Society Series A_ *98*, 39-54.

Fisher, R. A. (1962) Confidence limits for a cross-product ratio. _Australian Journal of Statistics_ *4*, 41.

Fisher, R. A. (1970) _Statistical Methods for Research Workers._ Oliver & Boyd.

Mehta, C. R. and Patel, N. R. (1986) Algorithm 643. FEXACT: A Fortran subroutine for Fisher's exact test on unordered r*c contingency tables. _ACM Transactions on Mathematical Software_, *12*, 154-161.

Clarkson, D. B., Fan, Y. and Joe, H. (1993) A Remark on Algorithm 643: FEXACT: An Algorithm for Performing Fisher's Exact Test in r x c Contingency Tables. _ACM Transactions on Mathematical Software_, *19*, 484-488.

Patefield, W. M. (1981) Algorithm AS159. An efficient method of generating r x c tables with given row and column totals. _Applied Statistics_ *30*, 91-97.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), Categorical Data Analysis Using the SAS System, Second Edition, Cary, NC: SAS Institute Inc.

See Also

`fisher.test`, `chisq.test`

Examples

```
#  library(bstats)
x = c(1,0)
n = c(72370,73058)
oddsratio(x,n=n)

Convictions <-
     matrix(c(2, 10, 15, 3),
            nrow = 2,
            dimnames =
            list(c("Dizygotic", "Monozygotic"),
                 c("Convicted", "Not convicted")))
     Convictions
     fisher.test(Convictions, conf.level = 0.95)$conf.int

x = matrix(c(2,10,17,13), ncol=2)
oddsratio(x)

Convictions <-
     matrix(c(8, 492, 0, 500), nrow = 2, byrow=TRUE)
fisher.test(Convictions, conf.level = 0.95)$conf.int

x = c(8,0)
n = c(500,500)
oddsratio(x,n=n)
```

ofc *occipitofrontal head circumference data*

Description

OFC data for singleton live births with gestational age at least 38 weeks.

Usage

```
data(ofc)
```

Format

A data frame with 2019 observations on 4 variables.

`Year`	numeric	2006 – 2009
`Sex`	character	’male’ or ’female’
`Gestation`	numeric	Gestational age (in weeks).
`Head`	numeric	head size.

References

Wang, B and Wertelecki, W, CSDA paper.

---

perm.test.NGS *A permutation test for DEG detection*

Description

To perform a permutation test to check whether a gene is differentially expressed.

Usage

```
perm.test.NGS(x,y, alternative = "two.sided", iter = 1001)
```

Arguments

`x,y`	The expressions of a gene in two populations
`alternative`	Type of test
`iter`	Iterations for the test.

Author(s)

B. Wang <bwang@southalabama.edu>

Examples

```
# To be updated.
```

---

ppower *Compute the power of an ANOVA-type test for proportions*

Description

To compute the power of an ANOVA-type test for proportions. It can be used for the sample size estimation.

Usage

```
ppower(effect.size, ngroup, n0, n1, range.p, alpha=0.05, gridsize=512)
```

Arguments

`effect.size`	Difference between the two proportions
`ngroup`	Levels of the treatment
`n0, n1`	Sample sizes.
`range.p`	Approximate range of the proportion of control group.
`alpha`	Significance level: experimentwise error rate.
`gridsize`	A (fine) grid size to compute the power when no (less) prior knowledge about `p0` can be used.

Value

The power of the test.

References

Wang, Bin. (TBA).

`robot` *Robot surgery.*

Description

Robot surgery results.

Usage

```
data(robot)
```

Format

A data frame with 388 rows and 111 columns (variables).

References

de Leval, M. R., Francois, K., Bull, C., Brawn, W., Spiegelhalter, D., Mar 1994. Analysis of a cluster of surgical failures. Application to a series of neonatal arterial switch operations. J. Thorac. Cardiovasc. Surg. 107, 914–923.

Page, E. S., 1954. Continuous inspection schemes. Biometrika 41 (1/2), pp. 100–115. URL http://www.jstor.org/stable/2333009

Parsonnet, V., Dean, D., Bernstein, A. D., Jun 1989. A method of uniform stratification of risk for evaluating the results of surgery in acquired adult heart disease. Circulation 79, 3–12.

Steiner, S. H., Cook, R. J., Farewell, V. T., Jan 1999. Monitoring paired binary surgical outcomes using cumulative sum charts. Stat Med 18, 69–86.

Steiner, S. H., Cook, R. J., Farewell, V. T., Treasure, T., Dec 2000. Monitoring surgical performance using risk-adjusted cumulative sum charts. Biostatistics 1, 441–452.

See Also

`cusum`

Examples

```
data(robot)
dim(robot)
```

roc *Receiver operating characteristic (ROC)*

Description

To compute and draw a ROC curve.

Usage

```
roc(y, x, z=NULL, event=”less”)
```

Arguments

`y` Logical. Actual status: event or not.

`x` Numberical vector based on which the diagnosis is performed.

`z` Logical. A covariate for diagnosis in addition to ’x’.

`event` Default: `less`. To classify a case to be an `event` if `x` is less than a `cutoff`. Otherwise, a case will be classified as an event of if `x` is greater than a `cutoff`.

Value

The TPR or sensitivity, and FPR or (1-specificity) will be returned together with the cutoffs.

The area under the ROC curve can be computed using `summary` function.

The ROC curve can be ploted using `plot` function.

Author(s)

B. Wang <`bwang@southalabama.edu`>

Examples

```
x <- rnorm(100)
y <- rnorm(100)<0
out = roc(y,x)
```

zr *occipitofrontal head circumference data*

Description

OFC data for singleton live births with gestational age at least 38 weeks in city "R" and county "Z".

Usage

```
data(zr)
```

Format

A data frame with three variables.

`Sex`	character	'male' or 'female'
`Region`	character	city "R" or county "Z".
`Head`	numeric	head size.

References

Wang, B. and Wertelecki, W. (2012) Density Estimation for Data With Rounding Errors. *Computational Statistics and Data Analysis*, (in press), doi: 10.1016/j.csda.2012.02.016.

Examples

```
data(zr)
zr
```

Index