Package ‘emplikAUC’

January 13, 2023

Version 0.3

Title Empirical Likelihood Ratio Test/Confidence Interval for AUC or pAUC

Maintainer Mai Zhou <maizhou@gmail.com>

Depends R (>= 3.2.5), emplik2, rootSolve

Imports stats

Description Test hypotheses and construct confidence intervals for AUC (area under Receiver Operating Characteristic curve) and pAUC (partial area under ROC curve), from the given two samples of test data with disease/healthy subjects. The method used is based on two sample empirical likelihood, as described in <https://www.ms.uky.edu/~mai/research/eAUC1.pdf>.

NeedsCompilation no

Author Mai Zhou [aut, cre, cph],
Xue Ding [aut]

Repository CRAN

Date/Publication 2023-01-13 19:20:02 UTC

R topics documented:

`EL2paucT1T2` *Testing one pAUC(p1, p2) and two quantile values together by Empirical Likelihood.*

Description

This function computes the two sample Log Empirical Likelihood ratio for testing H_0: pAUC(p1, p2) = theta; F(tau1)=1-p2; F(tau2)= 1-p1. The two samples are in the x-vector and y-vector.

Usage

```
EL2paucT1T2(tauVec, pauc, partial1, partial2, x, y, epsxy)
```

Arguments

`tauVec`	The vector (tau1, tau2), the two quantile values, to be tested.
`pauc`	The value of the pAUC(p1, p2) under H_0, to be tested.
`partial1`	The probability that define the quantile 1
`partial2`	The probability that define the quantile 2. Must satisfy partial 1 < partial 2 .
`x`	a vector of observations, length m, for the first sample.
`y`	a vector of observations, length n, for the second sample.
`epsxy`	The parameter for smoothing when compare x-y.

Details

The input `tauVec=(tau1, tau2)`, and must have: tau1 < tau2. The relavant definitions are: tau1 = F^{-1}(1-partial2); tau2 = F^{-1}(1-partial1). Thus, we must have partial2 > partial1.

This function is testing 3 parameters simultanuously. It depend on the package `emplik2`.

The empirical likelihood we used here is defined as

$$EL = \prod_{i=1}^{m} v_i \prod_{j=1}^{n} \nu_j \; ; \quad s.t. \quad \sum v_i = 1 \; , \; \sum \nu_j = 1 \; .$$

Value

A single value that is the "-2LLR" from `emplik2::el2.cen.EMm()`. Typically should be distributed as chi square df=3, under H_0.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

`el2test4auc` *Testing one AUC value by Empirical likelihood.*

Description

By calling upon a function from the package `emplik2` (using EM), this function computes the two sample Log Empirical Likelihood ratio for testing H_0: AUC = theta. The two samples are in the x-vector and y-vector inputs.

Usage

```
el2test4auc(theta, x, y, ind)
```

Arguments

`theta`	The "true" value of the AUC under H_0, to be tested.
`x`	a vector of observations, length m, for the first sample. The test-results of healthy subjects.
`y`	a vector of observations, length n, for the second sample. The test-results of desease subjects.
`ind`	A smoothed indicator function, to generate a Matrix of (smoothed) indicator values: I[x[i] < y[j]].

Details

This function is similar to the function `eltest4aucONE()`, the difference is that we call the function `emplik2::el2.cen.EMs()` to do the heavy computation (instead of by our own code). So, the speed and convergence property may be slightly different. When they both converge the results should be the same.

The empirical likelihood we used here is defined as

$$EL = \prod_{i=1}^{m} v_i \prod_{j=1}^{n} \nu_j \ ; \qquad \sum v_i = 1 \ , \ \ \sum \nu_j = 1 \ .$$

Value

A list that is the same as el2.cen.EMs() from emplik2 package. Which contains

`lambda`	The final tilting parameter.
`"-2LLR"`	The -2 log empirical likelihood ratio.
`Pval`	The p-value.
`iterNum`	The iteration number used in computing.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
#### The estimation of AUC
sum(smooth3(x=x, y=y))/(length(x)*length(y))
#### This does not work in Rcmd check: (truncate at %*%)
####   rep(1/length(x), length(x))%*%smooth3(x=x, y=y)%*%rep(1/length(y), length(y))
#### The result should be 0.75.
#### We may then test a hypothesis about the AUC value: H0: AUC= 0.7
el2test4auc(theta=0.7, x=x, y=y, ind=smooth3)
#### Two of the outputs should be '-2LLR'=0.1379561 and Pval=0.7103214
```

`el2testPauc` *Testing one pAUC(0, p) value by Empirical likelihood.*

Description

This function computes the two sample Log Empirical Likelihood ratio for testing H_0: pAUC(0, p) = theta. The two samples are in the x-vector and y-vector inputs.

Usage

```
el2testPauc(theta, x, y, ind, nuilow, nuiup, partial, epsxy, epsT)
```

Arguments

`theta`	The "true" value of the pAUC(0, p) under H_0, to be tested.
`x`	a vector of observations, length m, for the first sample, test-results with the healthy subjects.
`y`	a vector of observations, length n, for the second sample, test-results with the desease subjects.
`ind`	The (smoothed) indicator function for compare x-y.
`nuilow`	Lower bound for the nuisamce parameter (1-p)-th quantile of X) search.
`nuiup`	Upper bound for nuisance parameter search.
`partial`	The probability p in pAUC(0, p).
`epsxy`	The smoothing parameter when compare x-y.
`epsT`	The smoothing parameter when calculating quantile.

Details

This function will call another function: `el2testPaucT( )`. We then use `optimize( )` to profile out the nuisance parameter tau: the (1-p)-th quantile of X distribution.

Can be used by `findUnew( )` etc.

The empirical likelihood we used here is defined as

$$EL = \prod_{i=1}^{m} v_i \prod_{j=1}^{n} \nu_j \; ; \quad s.t. \quad \sum v_i = 1 \; , \quad \sum \nu_j = 1 \; .$$

Value

A list containing

`"-2LLR"`	The -2 log empirical likelihood ratio.
`Nupar`	The nuisance parameter value that achieved the minimum.
`Pval`	The p-value, by using chi square distribution with 1 df.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

`el2testPaucT` *Testing one pAUC(0, p) value and one quantile: F(tau) = 1-p together by Empirical Likelihood.*

Description

This function computes the two sample Log Empirical Likelihood ratio for testing H_0: pAUC(0, p) = theta and F(tau) = 1-p. (F is the CDF of X). The two samples data are in the x-vector and y-vector inputs. It uses EM.

Usage

```
el2testPaucT(tau, pauc, ind, partial, x, y, epsxy, epsT)
```

Arguments

`tau`	The "true" value of the (1-p)-th quantile of X-distribution F, to be tested.
`pauc`	The H_0 value of pAUC(0, p) to be tested.
`ind`	A smoothed indicator function, to generate a Matrix of (smoothed) indicator values: I[x[i] < y[j]].
`partial`	The probability p in pAUC(0, p); also the p in F(tau) = 1-p.
`x`	a vector of observations, length m, for the first sample, test-results with healthy subjects.
`y`	a vector of observations, length n, for the second sample, test-results with desease subjects.
`epsxy`	The smoothing parameter when compare x-y.
`epsT`	The smoothing parameter when compare x to Tau, for quantile estimation.

Details

This function is called by `el2testPauc( )`. It is listed here stand alone because users may find it useful elsewhere.

It make use of function `smooth3( )` and the function `el2.cen.EMm( )` from the `emplik2` package.

The empirical likelihood we used here is defined as

$$EL = \prod_{i=1}^{m} v_i \prod_{j=1}^{n} \nu_j \ ; \qquad \sum v_i = 1 \ , \ \ \sum \nu_j = 1 \ .$$

Value

It returns one value that is the "-2LLR". It should have chi square df=2 under H_0.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

`eltest4aucONE` *Testing one AUC value by Empirical likelihood.*

Description

This function computes the two sample Log Empirical Likelihood ratio for testing H_0: AUC = theta. The two samples are in the x-vector and y-vector.

Usage

```
eltest4aucONE(theta, x, y, ind, tol.u, tol.v, tol.H0)
```

Arguments

`theta`	The "true" value of the AUC under H_0, to be tested.
`x`	a vector of observations, length m, for the first sample. The test-results of healthy subjects
`y`	a vector of observations, length n, for the second sample. The test-results of desease subjects.
`ind`	A smoothed indicator function, to generate a Matrix of (smoothed) indicator values: I[x[i] < y[j]].
`tol.u`	Error tol for final u probability vector. Must > 0.
`tol.v`	Error tol for final v probability vector. Must > 0.
`tol.H0`	The error bound for the constrained NPMLE to satisfy H_0, must >0.

Details

This function is similar to `el2test4auc`, but using our own algorithm (not EM). It may be slightly different to the above in terms of speed and convergence property. We listed 3 kind of tol to control convergence.

The empirical likelihood we used here is defined as

$$EL = \prod_{i=1}^{m} v_i \prod_{j=1}^{n} \nu_j \; ; \qquad \sum v_i = 1 \; , \quad \sum \nu_j = 1 \; .$$

Value

A list containing

lambda	The final tilting parameter.
u	the new u vector.
v	The new v vector.
"-2LLR"	The -2 log empirical likelihood ratio.
Pval	The p-value.
iterNum	The iteration number used in computing.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
#### We know the AUC estimator here is 0.75.
#### We may test a hypothesis about the AUC value: H0: AUC= 0.7
eltest4aucONE(theta=0.7, x=x, y=y, ind=smooth3, tol.u=1e-6, tol.v=1e-6, tol.H0=1e-6)
#### Two of the outputs should be '-2LLR'=0.1379561 and Pval=0.7103214
```

---

eltest4paucONE *Testing one pAUC value by Empirical likelihood.*

Description

This function computes the two sample Log Empirical Likelihood ratio for testing H_0: pAUC(0, p) = theta. The two samples are in the x-vector and y-vector.

Usage

```
eltest4paucONE(theta,x,y,nuilow,nuiup,ind,partial,epsxy=0.05,epsT=(length(x))^(-0.75))
```

Arguments

`theta`	The "true" value of the pAUC under H_0, to be tested.
`x`	a vector of observations, length m, for the first sample. Test-results with healthy subjects.
`y`	a vector of observations, length n, for the second sample. Test-results with desease subjects.
`nuilow`	The lower bound for the nuisance parameter (the (1-p)-th quantile of X CDF F) search.
`nuiup`	The upper bound for the nuisance parameter search.
`ind`	A smoothed indicator function, to generate a Matrix of (smoothed) indicator values: I[x[i] < y[j]].
`partial`	The probability p in the pAUC(0,p).
`epsxy`	Window width for the smoother, "ind", when compare x-y.
`epsT`	Window width for the smoother, "ind", when define quantile.

Details

This function calls the function `eltest4paucT`. We then use `optimize( )` to profile out the nuisance parameter. Return an empirical likelihood ratio siutable for testing one parameter pAUC(0,p).

The empirical likelihood we used here is defined as

$$EL = \prod_{i=1}^{m} v_i \prod_{j=1}^{n} \nu_j \; ; \qquad \sum v_i = 1 \; , \quad \sum \nu_j = 1 \; .$$

Value

A list containing

`"-2LLR"`	The -2 log empirical likelihood ratio.
`Nupar`	The nuisance parameter value to achieve the minimum.
`Pval`	The p-value.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

`eltest4paucT` *Testing one pAUC and one quantile together by Empirical Likelihood.*

Description

This function computes the two sample Log Empirical Likelihood ratio for testing H_0: pAUC(0,p) = theta and F(tau) = 1-p. The two samples are in the x-vector and y-vector.

Usage

```
eltest4paucT(tau, x, y, true, ind, epsxy, epsT, tol.u, tol.v, tol.H0, p)
```

Arguments

`tau`	The "true" value of the (1-p)th quantile of X, under H_0, to be tested.
`x`	a vector of observations, length m, for the first sample. Test-results with healthy subjects.
`y`	a vector of observations, length n, for the second sample. Test-results with desease subjects.
`true`	The H_0 value of the pAUC(0, p) to be tested.
`ind`	A smoothed indicator function, to generate a Matrix of (smoothed) indicator values: I[x[i] < y[j]].
`epsxy`	Window width for smoother (ind) when compare x-y.
`epsT`	Window width for smoother (ind) when find quantile.
`tol.u`	Error tol for final u probability vector. Must > 0.
`tol.v`	Error tol for final v probability vector. Must > 0.
`tol.H0`	The error bound for checking if the constrained NPMLE satisfy H0, must >0.
`p`	The probability p in pAUC(0, p), and also in F(tau) = 1-p.

Details

This function is similar to `el2testPaucT( )`. Just a different algorithm (not EM). Speed and convergence may be slightly different.

This function is called by `eltest4paucONE`. It is listed here because the user may find it useful elsewhere.

The empirical likelihood we used here is defined as

$$EL = \prod_{i=1}^{m} v_i \prod_{j=1}^{n} \nu_j \; ; \qquad \sum v_i = 1 \; , \quad \sum \nu_j = 1 \; .$$

Value

A list containing

`lambda`	The final tilting parameter.
`u`	the new u vector.
`v`	The new v vector.
`"-2LLR"`	The -2 log empirical likelihood ratio.
`iterNum`	The iteration number used in computing.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

---

findLnew — *Finding the Lower bound of a confidence interval for theta by repeatedly testing the hypothesis for the parameter theta.*

Description

This function try to find the Lower bound of a confidence interval by repeatedly testing the hypothesis for the parameter theta until we reach the `level` for the "-2LLR" which return from the user supplied function `fun`.

Usage

```
findLnew(step=0.01, initStep=0, fun, MLE, level=qchisq(0.95,df=1),
            tol=.Machine$double.eps^0.5,...)
```

Arguments

`step`	Search step. Must > 0.
`initStep`	The initial step from MLE. May be used if we knew the bound is far away from MLE. This help to speed up things. Should be >=0.
`fun`	The function that should return "-2LLR".
`MLE`	The MLE of the parameter.

`level`	The level of the confidence. Default to 3.84 which is 95 percent confidence.
`tol`	The error bound for achieving the level given.
`...`	Any additional input to be passed to fun.

Details

This function is similar to `findUnew( )`.

Value

It returns a list containing

`Low`	Lower bound of the confidence interval.
`FstepL`	The error when search for Lower bound.
`Lvalue`	The final likelihood ratio value for Lower bound. Should =~ level.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

---

findULNEW — *Finding the Upper and Lower bound of a confidence interval for theta by repeatedly testing the hypothesis for the parameter theta.*

Description

This function try to find the Upper and Lower bound of a confidence interval by repeatedly testing the hypothesis for the parameter theta until we reach the given level for the "-2LLR" which is returned from the user supplied function `fun`.

Usage

```
findULNEW(step=0.01, initStep=0, fun, MLE, level=qchisq(0.95,df=1),
          tol=.Machine$double.eps^0.5,...)
```

Arguments

`step`	Search step. Must > 0.
`initStep`	The initial step from MLE. May be used if we knew the bound is far away from MLE. This help to speed up things. Should be >=0.
`fun`	The function that should return "-2LLR".
`MLE`	The MLE of the parameter.
`level`	The level of the confidence. Default to 3.84 which is 95 percent confidence (assume df=1).
`tol`	The error bound for achieving the level given.
`...`	Any additional input to be passed to fun.

Details

This function just combines the two functions `findUnew( )` and `findLnew( )`.

Value

It returns a list containing

`Low`	Lower bound of the confidence interval.
`Up`	Upper bound of the confidence interval.
`FstepL`	The error when search for Lower bound.
`FstepU`	The error when search for Upper bound.
`Lvalue`	The final likelihood ratio value for Lower bound. Should = level.
`Uvalue`	The final likelihood ratio value for Upper bound. Should = level.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

findUnew — *Finding the Upper bound of a confidence interval for* `theta` *by repeatedly testing the hypothesis for the parameter* `theta`.

Description

This function try to find the Upper bound of a confidence interval by repeatedly testing the hypothesis for the parameter theta until we reach the level for the "-2LLR" which return from the user supplied function `fun`.

Usage

```
findUnew(step=0.01, initStep=0, fun, MLE, level=qchisq(0.95,df=1),
          tol=.Machine$double.eps^0.5,...)
```

Arguments

`step`	Search step. Must > 0.
`initStep`	The initial step from MLE. May be used if we knew the bound is far away from MLE. This help to speed up things. Should be >=0.
`fun`	The user supplied function that should return "-2LLR".
`MLE`	The MLE of the parameter. An approximate value should be OK.
`level`	The level of the confidence. Default to 3.84 which is 95 percent confidence.
`tol`	The error bound for achieving the level given.
`...`	Any additional input to be passed to fun.

Details

This function is similar to `findLnew( )`.

The search is separate (for upper and lower) since Upper and Lower bound may behave differently and require different nuisance parameters inputs (in `...`).

Value

It returns a list containing

`Up`	Upper bound of the confidence interval.
`FstepU`	The error when search for Upper bound.
`Uvalue`	The final likelihood ratio value for Upper bound. Should =~ level.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

`myEstPaucT` *Given the x, y 2-sample data, first estimate the (1-partial)-th quantile of X sample, then estimate the pAUC(0, partial), with the plug-in estimated quantile.*

Description

This function computes the estimator using smoothed Indicator function I[x < y] which is a 3rd order polynomial.

eps is a scalar, must > 0. It is the smoothing window width for indicator function when compare x-y. $epsT$ is a scalar, must > 0. It is the smoothing window width for estimating quantile function when compare x with tau.

Usage

```
myEstPaucT(x, y, partial, eps=0.05, epsT=(length(x))^(-0.75))
```

Arguments

`x`	a vector of observations, length m, for the first sample. Test-results with healthy subjects.
`y`	a vector of observations, length n, for the second sample. Test-results with desease subjects.
`partial`	The probability in pAUC(0, partial).
`eps`	The smoothing window width, for indicator I[x < y]. Must >0.
`epsT`	The smoothing window width for the quantile estimation. Must >0.

Details

This function gives the estimators that are consistant with our smoothing used in the computation of empirical likelihood. Typically the smoother for quantile should be a bit "smoother" that usual.

Known problem: when input `partial` is too close to 1 or 0, it will fail. When `partial` is equal to 1, then the pAUC reduces to AUC, which can be estimated easily, as the example below shows.

Why not also give the AUC estimator? May be it is too easy. See example below.

Value

This function returns a list, with

`"tau(1-partial)"`
: The estimator for the (1-partial)-th quantile of X-distribution,

and

`"Pauc(0, partial)"`
: The estimated pAUC.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
#### To get an estimator of the AUC (not pAUC), using our smooth3( ), we do
sum(smooth3(x=x, y=y))/(length(x)*length(y))
#### We should get AUC = 0.75.
#### To get the estimator of the pAUC(0, 0.3) AND the 70th quantile of x sample
myEstPaucT(x=x, y=y, partial=0.3)
#### We should get estimator tau(1-0.3)=239.9474 and pAUC(0, 0.3)=0.1416667.
```

---

quantONE — *Smoothed quantile estimation from the given x-sample.*

Description

This function computes the smoothed quantile estimate, using the smoothing function `smooth3`. (or it can be changed easily) Otherwise it is similar to R function `quantile()`. Compare to `quantile(x, prob, type=9)`.

$myeps$ is a scalar, must > 0. It is the smoothing window width.

Usage

```
quantONE(x, prob, myeps=(length(x))^(-0.75))
```

Arguments

x	a vector of observations, length m, the sample.
prob	a probabilty.
myeps	The smoothing window width, must >0.

Details

This function is called by myEstPaucT. It is listed here because the user may find it useful elsewhere.

Known problems: when the input prob is too close to 0 or 1 (distance less than 0.03), the computation will stop. Also, if the solution to the equation

$$\tau \mid \quad \frac{1}{m}\sum_{i=1}^{m} smooth3vec(X_i, \tau, myeps) = prob$$

is not unique, this function only returns one of the solutions.

Value

The function quantONE returns a scalar that is the estimated (prob)-th quantile of X distribution.

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
#### To estimate the 70-th percentile of x distrubution:
quantONE(x=x, prob=0.7)
#### we should get 239.9474.
```

---

smooth3 — *Smoothed indicator function I[x < y], which is the integration of the Epanechnikov kernal.*

Description

This function computes the smoothed Indicator function I[x < y] using a 3rd order polynomial.

If |x-y| > eps then the result is the same as the indicator function I[x < y] (either 0 or 1). For |x-y| < eps, it is a 3rd order polynomial.

eps is a scalar, must > 0. It is the smoothing window width.

Usage

```
smooth3(x, y, eps=0.05)
```

Arguments

x	a vector of observations, length m, for the first sample.
y	a vector of observations, length n, for the second sample.
eps	The smoothing window width, must >0.

Details

This function is used in many places to replace an indicator function $I[x < y]$. For example, when estimating the AUC. It is listed here because users may find it useful elsewhere.

Value

`smooth3( )` returns a matrix of dimension `ncol=length(y)`, `nrow=length(x)`. The entry of the matrix are smoothed values of I[x[i] < y[j]].

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

---

smooth3vec — *Smoothed indicator function I[x < const], which is the integration of the Epanechnikov kernal.*

Description

This function computes the smoothed Indicator function I[x < const] using a 3rd order polynomial.

If |x - const| > eps then the result is the same as the indicator function I[x < const] (either 0 or 1). For |x - const| < eps, it is a 3rd order polynomial.

eps is a scalar, must > 0. It is the smoothing window width.

Usage

```
smooth3vec(x, const, eps=0.05)
```

Arguments

x	a vector of observations, length m, for the first sample.
const	a single number.
eps	The smoothing window width, must be >0. Ideally this should be sample size dependent.

Details

This function is similar to `smooth3` but only compare the `x` vector to a single number and thus returns a vector instead of matrix. You may also use the `smooth3()` with a bit care, for that matter, but this vector version should be faster and save memory. It is listed here because the user may find it useful elsewhere.

We used this function to estimate the quantile from the x-sample.

Value

`smooth3vec` returns a vector of length=length(x). The entry of the matrix are smoothed values of I[x[i] < const].

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

smooth5vec — *Smoothed indicator function I[x < const], which is the integration of the Quartic kernal.*

Description

This function computes the smoothed Indicator function I[x < const] using a 5th order polynomial.

If |x - const| > eps then the result is the same as the indicator function I[x < const] (either 0 or 1). For |x - const| < eps, it is a 5th order polynomial.

eps is a scalar, must > 0. It is the smoothing window width.

Usage

```
smooth5vec(x, const, eps=0.05)
```

Arguments

`x`	a vector of observations, length m, for the first sample.
`const`	a single number.
`eps`	The smoothing window width, must be >0. Ideally, this should be sample size dependent.

Details

This function is twice continuously differenciable, smoother than `smooth3vec`. It is listed here because the user may need extra smoothness (compare to `smooth3vec`) and may find it useful elsewhere.

Value

`smooth5vec` returns a vector of length=length(x). The entry of the vector are smoothed values of I[x[i] < const].

Author(s)

Mai Zhou <maizhou@gmail.com>.

References

Zhao, Y., Ding, X. and Zhou (2021). Confidence Intervals of AUC and pAUC by Empirical Likelihood. Tech Report. https://www.ms.uky.edu/~mai/research/eAUC1.pdf

Examples

```
y <- c(10, 209, 273, 279, 324, 391, 566, 785)
x <- c(21, 38, 39, 51, 77, 185, 240, 289, 524)
```

Index