Package ‘WWR’

October 12, 2022

Type Package

Version 1.2.2

Date 2017-10-24

Title Weighted Win Loss Statistics and their Variances

Description
Calculate the (weighted) win loss statistics including the win ratio, win difference and win product
and their variances, with which the p-
values are also calculated. The variance estimation is based on
Luo et al. (2015) <doi:10.1111/biom.12225> and Luo et al. (2017) <doi:10.1002/sim.7284>. This package also calculates general win loss statistics with
user-specified win loss function with variance estimation based on
Bebu and Lachin (2016) <doi:10.1093/biostatistics/kxv032>.
This version corrected an error when outputting confidence interval for win difference.

Depends R (>= 3.1.2)

Imports inline, stats

License GPL (>= 2)

RoxygenNote 5.0.1

LazyData true

NeedsCompilation yes

Author Xiaodong Luo [aut, cre],
Junshan Qiu [ctb],
Steven Bai [ctb],
Hong Tian [ctb],
Mike Mikailov [ctb],
Sanofi [cph]

Maintainer Xiaodong Luo <Xiaodong.Luo@sanofi.com>

Repository CRAN

Date/Publication 2017-10-25 03:40:18 UTC

R topics documented:

`WWR-package` *Weighted Win Loss Statistics and their Variances*

Description

Calculate the (weighted) win loss statistics including the win ratio, win difference and win product and their variances, with which the p-values are also calculated. The variance estimation is based on Luo et al. (2015) <doi:10.1111/biom.12225> and Luo et al. (2017) <doi:10.1002/sim.7284>. This package also calculates general win loss statistics with user-specified win loss function with variance estimation based on Bebu and Lachin (2016) <doi:10.1093/biostatistics/kxv032>. This version corrected an error when outputting confidence interval for win difference.

Details

The DESCRIPTION file:

Package:	WWR
Type:	Package
Version:	1.2.2
Date:	2017-10-24
Title:	Weighted Win Loss Statistics and their Variances
Description:	Calculate the (weighted) win loss statistics including the win ratio, win difference and win product and their
Authors@R:	c(person(given="Xiaodong", family="Luo", email = "Xiaodong.Luo@sanofi.com", role =c("aut", "cre")), pe
Depends:	R (>= 3.1.2)
Imports:	inline, stats
License:	GPL (>= 2)
RoxygenNote:	5.0.1
LazyData:	true
Author:	Xiaodong Luo [aut, cre], Junshan Qiu [ctb], Steven Bai [ctb], Hong Tian [ctb], Mike Mikailov [ctb], Sanofi [
Maintainer:	Xiaodong Luo <Xiaodong.Luo@sanofi.com>

Index of help topics:

```
WWR-package             Weighted Win Loss Statistics and their
                        Variances
genwr                   General Win Loss Statistics
winratio                Win Loss Statistics
```

wlogr2	Log-rank statistics
wwratio	Weighted Win Loss Statistics

Author(s)

NA

Maintainer: NA

References

Pocock S.J., Ariti C.A., Collier T. J. and Wang D. 2012. The win ratio: a new approach to the analysis of composite endpoints in clinical trials based on clinical priorities. European Heart Journal, 33, 176-182.

Luo X., Tian H., Mohanty S. and Tsai W.-Y. 2015. An alternative approach to confidence interval estimation for the win ratio statistic. Biometrics, 71, 139-145.

Bebu I. and Lachin J.M. 2016. Large sample inference for a win ratio analysis of a composite outcome based on prioritized components. Biostatistics, 17, 178-187.

Luo X., Qiu J., Bai S. and Tian H. 2017. Weighted win loss approach for analyzing prioritized outcomes. Statistics in Medicine, <doi: 10.1002/sim.7284>.

`genwr` *General Win Loss Statistics*

Description

Calculate the general win loss statistics and their corresponding variances under the global NULL hypothesis and under alterantive hypothesis based on Bebu and Lachin (2016) paper, which is a generalization of the win ratio of Pocock et al. (2012) and the win difference of Luo et al. (2015). This calculation needs the users to specify the win loss matrix.

Usage

```
genwr(aindex)
```

Arguments

`aindex` a numeric matrix of win loss indicators. Suppose there are group 1 and group 0 in the study with sample sizes n_1 and n_0 respectively. The matrix `aindex` is a $n_1 \times n_0$ matrix with elements $C_{ij} : i = 1, \ldots, n_1, j = 1, \ldots, n_0$. The element C_{ij} is equal to 1 if subject i in group 1 wins over subject j in group 0 on the most important outcome, C_{ij} is equal to -1 if subject i in group 1 loses against subject j in group 0 on the most important outcome; C_{ij} is equal to 2 if subject i in group 1 wins over subject j in group 0 on the second most important outcome after tie on the most important outcome, C_{ij} is equal to -2 if subject i in group 1 loses against subject j in group 0 on the second most important outcome after tie on the most important outcome; C_{ij} is equal to 3 if subject i in group 1 wins over subject j in group 0 on the third most important outcome after tie on the first two most important outcomes, C_{ij} is equal to -3 if subject i in group 1 loses against subject j in group 0 on the third most important outcome after tie on the first two most important outcomes; and so forth until all the outcomes have been used for comparison; then C_{ij} is equal to 0 if an ultimate tie is resulted.

Details

General win loss statistics

Value

`n1`	Number of subjects in group 1
`n0`	Number of subjects in group 0
`n`	Total number of subjects in both groups
`totalw`	Total number of wins in group 1
`totall`	Total number of losses in group 1
`tw`	A vector of total numbers of wins in group 1 for each of the outcomes. Note that `totalw`=sum(`tw`), the first element is for the most important outcome, the second elemnet is for the second important outcome etc.
`tl`	A vector of total numbers of losses in group 1 for each of the outcomes. Note that `totall`=sum(`tl`), the first element is for the most important outcome, the second elemnet is for the second important outcome etc.
`xp`	The ratios between `tw` and `tl`
`cwindex`	The win contribution index defined as the ratio between `tw` and `totalw+totall`
`clindex`	The loss contribution index defined as the ratio between `tl` and `totalw+totall`
`wr`	Win ratio defined as `totalw/totall`
`vr`	Asymptotic variance of the win ratio under alterantive hypothesis
`vr0`	Asymptotic variance of the win ratio under global null hypothesis
`tr`	standardized log(wr) using the variance `vr`
`pr`	2-sided p-value of `tr`
`tr0`	standardized log(wr) using the variance `vr0`
`pr0`	2-sided p-value of `tr0`
`wd`	Win difference defined as `totalw-totall`
`vd`	Asymptotic variance of the win difference under alterantive hypothesis. The first element is the variance when the group assignment is considered as fixed, and the second element is the variance when the group assignment is considered as random, so the second element is slightly larger than the first element when with unequal allocations.
`vd0`	Asymptotic variance of the win difference under global null hypothesis
`td`	standardized `wd` using the variance `vd`
`pd`	2-sided p-values of `td`

td0	standardized wd using the variance vd0
pd0	2-sided p-value of td0
wp	Win product defined as the product of tw/tl
vp	Asymptotic variance of the win product under alterantive hypothesis
vp0	Asymptotic variance of the win product under global null hypothesis
tp	standardized log(wp) using the variancevp
pp	2-sided p-value of tp
tp0	standardized log(wp) using the variancevp0
pp0	2-sided p-value of tp0

Author(s)

Xiaodong Luo

References

Pocock S.J., Ariti C.A., Collier T. J. and Wang D. 2012. The win ratio: a new approach to the analysis of composite endpoints in clinical trials based on clinical priorities. European Heart Journal, 33, 176-182.

Luo X., Tian H., Mohanty S. and Tsai W.-Y. 2015. An alternative approach to confidence interval estimation for the win ratio statistic. Biometrics, 71, 139-145.

Bebu I. and Lachin J.M. 2016. Large sample inference for a win ratio analysis of a composite outcome based on prioritized components. Biostatistics, 17, 178-187.

See Also

wlogr2,winratio,wwratio

Examples

```
##########################################################
## Example 1: survival (semi-competing risks) example
##            with terminal event having higher priority
##########################################################
##############################
##  Step 1: data generation
##############################
n<-200
rho<-0.5
b2<-0.0
b1<-0.0
bc<-1.0
lambda10<-0.1;lambda20<-0.08;lambdac0<-0.09
lam1<-rep(0,n);lam2<-rep(0,n);lamc<-rep(0,n)
z<-rep(0,n)
z[1:(n/2)]<-1

lam1<-lambda10*exp(-b1*z)
```

```
lam2<-lambda20*exp(-b2*z)
lamc<-lambdac0*exp(-bc*z)
tem<-matrix(0,ncol=3,nrow=n)

y2y<-matrix(0,nrow=n,ncol=3)
y2y[,1]<-rnorm(n);y2y[,3]<-rnorm(n)
y2y[,2]<-rho*y2y[,1]+sqrt(1-rho^2)*y2y[,3]
tem[,1]<--log(1-pnorm(y2y[,1]))/lam1
tem[,2]<--log(1-pnorm(y2y[,2]))/lam2
tem[,3]<--log(1-runif(n))/lamc

y1<-apply(tem,1,min)
y2<-apply(tem[,2:3],1,min)
d1<-as.numeric(tem[,1]<=y1)
d2<-as.numeric(tem[,2]<=y2)

###un-weighted win loss
wtest<-winratio(y1,y2,d1,d2,z)
summary(wtest)

i<-1 ##i=1,2,3,4
j<-1 ##j=1,2

###weighted win loss
wwtest<-wwratio(y1,y2,d1,d2,z,wty1=i,wty2=j)
summary(wwtest)

####general win loss
###Define the win loss function
comp<-function(y,x){
  y1i<-y[1];y2i<-y[2];d1i<-y[3];d2i<-y[4]
  y1j<-x[1];y2j<-x[2];d1j<-x[3];d2j<-x[4]
  w2<-0;w1<-0;l2<-0;l1<-0

  if (d2j==1 & y2i>=y2j) w2<-1
  else if (d2i==1 & y2j>=y2i) l2<-1

  if (w2==0 & l2==0 & d1j==1 & y1i>=y1j) w1<-1
  else if (w2==0 & l2==0 & d1i==1 & y1j>=y1i) l1<-1

  comp<-0
  if (w2==1) comp<-1
  else if (l2==1) comp<-(-1)
  else if (w1==1) comp<-2
  else if (l1==1) comp<-(-2)

  comp
}
###Use the user-defined win loss function to calculate the win loss matrix
y<-cbind(y1,y2,d1,d2)
yy1<-y[z==1,]
yy0<-y[z==0,]
n1<-sum(z==1)
```

```
n0<-sum(z==0)
bindex<-matrix(0,nrow=n1,ncol=n0)
for (i in 1:n1)for (j in 1:n0)bindex[i,j]<-comp(yy1[i,],yy0[j,])
###Use the calculated win loss matrix to calculate the general win loss statistics
bgwr<-genwr(bindex)
summary(bgwr)

###################################################################
# Note: if n>=1000 or the win loss function is complex,
#        one may experience long runtime. One may instead use C, C++,
#        Fortran, Python to code the win loss function.
#        The following provides an example using Fortran 95 code to
#        define the win loss matrix and then port it back to R
#        using the package "inline"
###################################################################

######################################################
#  This is to install and load package "inline"
#  so that we can compile user-defined
#  win loss function
#
#install.packages("inline")
library("inline")


###############################################################
#  You may also need to have rtools and gcc in the PATH
#  The following code add these
#  for the current R session ONLY
#  Please remove the '#' in the following 6 lines.
#
#rtools <- "C:\Rtools\bin"
#gcc <- "C:\Rtools\gcc-4.6.3\bin"
#path <- strsplit(Sys.getenv("PATH"), ";")[[1]]
#new_path <- c(rtools, gcc, path)
#new_path <- new_path[!duplicated(tolower(new_path))]
#Sys.setenv(PATH = paste(new_path, collapse = ";"))
###############################################################


###Define the win loss indicator by a user-supplied function
codex5 <- "
integer::i,j,indexij,d1i,d2i,d1j,d2j,w2,w1,l2,l1
double precision::y1i,y2i,y1j,y2j
do i=1,n1,1
   y1i=y(i,1);y2i=y(i,2);d1i=dnint(y(i,3));d2i=dnint(y(i,4))
   do j=1,n0,1
      y1j=x(j,1);y2j=x(j,2);d1j=dnint(x(j,3));d2j=dnint(x(j,4))
      w2=0;w1=0;l2=0;l1=0
      if (d2j==1 .and. y2i>=y2j) then
         w2=1
      else if (d2i==1 .and. y2j>=y2i) then
         l2=1
```

```
      end if

      if (w2==0 .and. l2==0 .and. d1j==1 .and. y1i>=y1j) then
         w1=1
      else if (w2==0 .and. l2==0 .and. d1i==1 .and. y1j>=y1i) then
         l1=1
      end if
      aindex(i,j)=0
      if (w2==1) then
         aindex(i,j)=1
      else if (l2==1) then
         aindex(i,j)=-1
      else if (w1==1) then
         aindex(i,j)=2
      else if (l1==1) then
         aindex(i,j)=-2
      end if
   end do
end do
"
###End of defining the win loss indicator by a user-supplied function

###Convert the above code to Fortran 95 code and port it back to R
cubefnx5<-cfunction(sig = signature(n1="integer",n0="integer",p="integer",
           y="numeric",x="numeric", aindex="integer"),
           implicit = "none", dim = c("", "", "", "(n1,p)","(n0,p)","(n1,n0)"),
           codex5, language="F95")

###Use the above defined function to calculate the win loss indicators
y<-cbind(y1,y2,d1,d2)
yy1<-y[z==1,]
yy0<-y[z==0,]
n1<-sum(z==1)
n0<-sum(z==0)
options(object.size=1.0E+10)
##The following is the win loss indicator matrix
aindex<-matrix(cubefnx5(n1,n0,length(y[1,]), yy1,yy0,
               matrix(0,nrow=n1,ncol=n0))$aindex,byrow=FALSE,ncol=n0)
###Use the win loss indicator matrix to calculate the general win loss statistics
agwr<-genwr(aindex)
summary(agwr)


##########################################################
## Example 2: Continuous outcome example
##  suppose there are two outcomes (y1,y2) following bivariate normal dist
##  y1 is more important than y2, when comparing with (x1,x2) from another subject
##  a win of first outcome if y1>x1+1 and a loss if y1<x1-1
##  if tie, i.e. |y1-x1|<=1, then a win of second outcome if y2>x2+0.5
##  and a loss if y2<x2-0.5. The other scenarios are tie.
##########################################################
##############################
##  Step 1: data generation
```

```
##############################
n<-300
rho<-0.5
b2<-2.5
b1<-2.0
z<-rep(0,n)
z[1:(n/2)]<-1

y2y<-matrix(0,nrow=n,ncol=3)
y2y[,1]<-rnorm(n);y2y[,3]<-rnorm(n)
y2y[,2]<-rho*y2y[,1]+sqrt(1-rho^2)*y2y[,3]
y1<-b1*z+y2y[,1]
y2<-b2*z+y2y[,2]


####general win loss

###Define the win loss indicator by a user-supplied function
codex6 <- "
integer::i,j,indexij,w2,w1,l2,l1
double precision::y1i,y2i,y1j,y2j
do i=1,n1,1
   y1i=y(i,1);y2i=y(i,2)
   do j=1,n0,1
      y1j=x(j,1);y2j=x(j,2)
      w2=0;w1=0;l2=0;l1=0
      if (y1i>(y1j+1.0)) then
         w1=1
      else if (y1i<(y1j-1.0)) then
         l1=1
      end if

      if (w1==0 .and. l1==0 .and. y2i>(y2j+0.5)) then
         w2=1
      else if (w1==0 .and. l1==0 .and. y2i<(y2j-0.5)) then
         l2=1
      end if
      aindex(i,j)=0
      if (w1==1) then
         aindex(i,j)=1
      else if (l1==1) then
         aindex(i,j)=-1
      else if (w2==1) then
         aindex(i,j)=2
      else if (l2==1) then
         aindex(i,j)=-2
      end if
   end do
end do
"
###End of defining the win loss indicator by a user-supplied function

###Convert the above code to Fortran 95 code and port it back to R
```

```
cubefnx6<-cfunction(sig = signature(n1="integer",n0="integer",p="integer",
           y="numeric",x="numeric", aindex="integer"),
           implicit = "none", dim = c("", "", "", "(n1,p)","(n0,p)","(n1,n0)"),
           codex6, language="F95")

###Use the above defined function to calculate the win loss indicators
y<-cbind(y1,y2)
yy1<-y[z==1,]
yy0<-y[z==0,]
n1<-sum(z==1)
n0<-sum(z==0)
options(object.size=1.0E+10)
##The following is the win loss indicator matrix
aindex<-matrix(cubefnx6(n1,n0,length(y[1,]), yy1,yy0,
               matrix(0,nrow=n1,ncol=n0))$aindex,byrow=FALSE,ncol=n0)

###Use the win loss indicator matrix to calculate the general win loss statistics
agwr<-genwr(aindex)
summary(agwr)
```

`winratio` *Win Loss Statistics*

Description

Calculate the win loss statistics of Pocock et al. (2012) and the corresponding variances, which are based on a U-statistic method of Luo et al. (2015)

Usage

```
winratio(y1,y2,d1,d2,z)
```

Arguments

`y1` a numeric vector of event times denoting the minimum of event times T_1, T_2 and censoring time C, where the endpoint T_2, corresponding to the terminal event, is considered of higher clinical importance than the endpoint T_1, corresponding to the non-terminal event. Note that the terminal event may censor the non-terminal event, resulting in informative censoring.

`y2` a numeric vector of event times denoting the minimum of event time T_2 and censoring time C. Clearly, y2 is not smaller than y1.

`d1` a numeric vector of event indicators with 1 denoting the non-terminal event is observed and 0 denoting otherwise.

`d2` a numeric vector of event indicators with 1 denoting the terminal event is observed and 0 denoting otherwise.Note that Luo et al. (2015) use a single indicator d so that $d = 1$ if and only if `d1=1` and `d2=1`; $d = 2$ if and only if `d1=0` and `d2=1`; $d = 3$ if and only if `d1=0` and `d2=0`; and $d = 4$ if and only if `d1=1` and `d2=0`.

`z`	a numeric vector of group indicators with 1 denoting the treatment group and 0 the control group.

Details

win loss statistics

Value

`n1`	Number of subjects in group 1
`n0`	Number of subjects in group 0
`n`	Total number of subjects in both groups
`totalw`	Total number of wins in group 1
`totall`	Total number of losses in group 1
`tw`	A vector of total numbers of wins in group 1 for each of the two outcomes. Note that `totalw`=sum(`tw`), and the first element is for the terminal event and the second element is for the non-terminal event.
`tl`	A vector of total numbers of losses in group 1 for each of the two outcomes. Note that `totall`=sum(`tl`), and the first element is for the terminal event and the second element is for the non-terminal event.
`xp`	The ratios between `tw` and `tl`
`cwindex`	The win contribution index defined as the ratio between `tw` and `totalw+totall`
`clindex`	The loss contribution index defined as the ratio between `tl` and `totalw+totall`
`wr`	win ratio
`vr`	estimated variance of win ratio
`tr`	standardized log(wr)
`pr`	2-sided p-value of tr
`wd`	win difference
`vd`	estimated variance of win difference
`td`	standardized wd
`pd`	2-sided p-value of td
`wp`	win product
`vp`	estimated variance of win product
`tp`	standardized log(wp)
`pp`	2-sided p-value of tp

Author(s)

Xiaodong Luo

References

Pocock S.J., Ariti C.A., Collier T. J. and Wang D. 2012. The win ratio: a new approach to the analysis of composite endpoints in clinical trials based on clinical priorities. European Heart Journal, 33, 176-182.

Luo X., Tian H., Mohanty S. and Tsai W.-Y. 2015. An alternative approach to confidence interval estimation for the win ratio statistic. Biometrics, 71, 139-145.

See Also

wlogr2,wwratio

Examples

```
n<-300
rho<-0.5
b2<-0.2
b1<-0.5
bc<-1.0
lambda10<-0.1;lambda20<-0.08;lambdac0<-0.09
lam1<-rep(0,n);lam2<-rep(0,n);lamc<-rep(0,n)
z<-rep(0,n)
z[1:(n/2)]<-1

lam1<-lambda10*exp(-b1*z)
lam2<-lambda20*exp(-b2*z)
lamc<-lambdac0*exp(-bc*z)
tem<-matrix(0,ncol=3,nrow=n)

y2y<-matrix(0,nrow=n,ncol=3)
y2y[,1]<-rnorm(n);y2y[,3]<-rnorm(n)
y2y[,2]<-rho*y2y[,1]+sqrt(1-rho^2)*y2y[,3]
tem[,1]<--log(1-pnorm(y2y[,1]))/lam1
tem[,2]<--log(1-pnorm(y2y[,2]))/lam2
tem[,3]<--log(1-runif(n))/lamc

y1<-apply(tem,1,min)
y2<-apply(tem[,2:3],1,min)
d1<-as.numeric(tem[,1]<=y1)
d2<-as.numeric(tem[,2]<=y2)
wtest<-winratio(y1,y2,d1,d2,z)
summary(wtest)
```

---

wlogr2 *Log-rank statistics*

Description

This will calculate the log-rank and Gehan statistics along with their variances

Usage

```
wlogr2(y, d, z, wty = 1)
```

Arguments

`y`	a vector of observed event times
`d`	a vector of event indicators with 1=event and 0=censored
`z`	a vector of group indicators with 1=treatment and 0=control
`wty`	a vector of weight indicators with 1=Gehan and 2=log-rank

Value

`wty`	Type of statistics, 1=Gehan, 2=log-rank
`stat`	value of the stat
`vstat`	estimated variance
`tstat`	standardized test stat
`pstat`	2-sided p-value of the standardized test stat

Note

This provides Gehan test that is usually ignored

Author(s)

Xiaodong Luo

References

Gehan E.A. 1965. A generalized Wilcoxon test for comparing arbitrarily single-censored samples. Biometrika, 53, 203-223.

Peto R. and Peto J. 1972. Asymptotically Efficient Rank Invariant Test Procedures. Journal of the Royal Statistical Society, Series A, 135, 185-207.

See Also

`winratio`,`wwratio`

Examples

```
n<-300
b<-0.2
bc<-1.0
lambda0<-0.1;lambdac0<-0.09
lam<-rep(0,n);lamc<-rep(0,n)
z<-rep(0,n)
z[1:(n/2)]<-1

lam<-lambda0*exp(-b*z)
```

```
lamc<-lambdac0*exp(-bc*z)
tem<-matrix(0,ncol=2,nrow=n)

tem[,1]<--log(1-runif(n))/lam
tem[,2]<--log(1-runif(n))/lamc

y<-apply(tem,1,min)
d<-as.numeric(tem[,1]<=y)

i<-1 ##i=1,2
wtest<-wlogr2(y,d,z,wty=i)
wtest
```

---

wwratio *Weighted Win Loss Statistics*

Description

Calculate weighted win loss statistics and their corresponding variances under the global NULL hypothesis based on Luo et al. (2017) paper, which is a generalization of the win ratio of Pocock et al. (2012) and the win difference of Luo et al. (2015)

Usage

```
wwratio(y1, y2, d1, d2, z, wty1 = 1, wty2 = 1)
```

Arguments

y1	a numeric vector of event times denoting the minimum of event times T_1, T_2 and censoring time C, where the endpoint T_2, corresponding to the terminal event, is considered of higher clinical importance than the endpoint T_1, corresponding to the non-terminal event. Note that the terminal event may censor the non-terminal event, resulting in informative censoring.
y2	a numeric vector of event times denoting the minimum of event time T_2 and censoring time C. Clearly, y2 is not smaller than y1.
d1	a numeric vector of event indicators with 1 denoting the non-terminal event is observed and 0 else.
d2	a numeric vector of event indicators with 1 denoting the terminal event is observed and 0 else.
z	a numeric vector of group indicators with 1 denoting the treatment group and 0 the control group.
wty1	a numeric vector of weight indicators for the non-terminal event with values 1 to 4 corresponding to weights used in Luo et al. (2017).
wty2	a numeric vector of weight indicators for the terminal event with values 1 to 2 corresponding to weights used in Luo et al. (2017).

Details

weighted win statistics

Value

n1	Number of subjects in group 1
n0	Number of subjects in group 0
n	Total number of subjects in both groups
wty1	Weight for non-terminal event
wty2	Weight for terminal event
totalw	Total number of wins in group 1
totall	Total number of losses in group 1
tw	A vector of total numbers of wins in group 1 for each of the two outcomes. Note that totalw=sum(tw), and the first element is for the terminal event and the second element is for the non-terminal event.
tl	A vector of total numbers of losses in group 1 for each of the two outcomes. Note that totall=sum(tl), and the first element is for the terminal event and the second element is for the non-terminal event.
xp	The ratios between tw and tl
cwindex	The win contribution index defined as the ratio between tw and totalw+totall
clindex	The loss contribution index defined as the ratio between tl and totalw+totall
wr	weighted win ratio
vr	estimated variance of weighted win ratio
tr	standardized log(wr)
pr	2-sided p-value of tr
wd	weighted win difference
vd	estimated variance of weighted win difference
td	standardized wd
pd	2-sided p-value of td
wp	weighted win product
vp	estimated variance of weighted win product
tp	standardized log(wp)
pp	2-sided p-value of tp

Author(s)

Xiaodong Luo

References

Pocock S.J., Ariti C.A., Collier T. J. and Wang D. 2012. The win ratio: a new approach to the analysis of composite endpoints in clinical trials based on clinical priorities. European Heart Journal, 33, 176-182.

Luo X., Tian H., Mohanty S. and Tsai W.-Y. 2015. An alternative approach to confidence interval estimation for the win ratio statistic. Biometrics, 71, 139-145.

Luo X., Qiu J., Bai S. and Tian H. 2017. Weighted win loss approach for analyzing prioritized outcomes. Statistics in Medicine, doi: 10.1002/sim.7284.

See Also

wlogr2,winratio

Examples

```
n<-300
rho<-0.5
b2<-0.2
b1<-0.5
bc<-1.0
lambda10<-0.1;lambda20<-0.08;lambdac0<-0.09
lam1<-rep(0,n);lam2<-rep(0,n);lamc<-rep(0,n)
z<-rep(0,n)
z[1:(n/2)]<-1

lam1<-lambda10*exp(-b1*z)
lam2<-lambda20*exp(-b2*z)
lamc<-lambdac0*exp(-bc*z)
tem<-matrix(0,ncol=3,nrow=n)

y2y<-matrix(0,nrow=n,ncol=3)
y2y[,1]<-rnorm(n);y2y[,3]<-rnorm(n)
y2y[,2]<-rho*y2y[,1]+sqrt(1-rho^2)*y2y[,3]
tem[,1]<--log(1-pnorm(y2y[,1]))/lam1
tem[,2]<--log(1-pnorm(y2y[,2]))/lam2
tem[,3]<--log(1-runif(n))/lamc

y1<-apply(tem,1,min)
y2<-apply(tem[,2:3],1,min)
d1<-as.numeric(tem[,1]<=y1)
d2<-as.numeric(tem[,2]<=y2)

i<-1 ##i=1,2,3,4
j<-2 ##j=1,2

wtest<-wwratio(y1,y2,d1,d2,z,wty1=i,wty2=j)
summary(wtest)
```

Index