Package 'WINS'

July 18, 2022

Type Package

Title The R WINS Package

Version 1.2

Maintainer Ying Cui <cuiyingbeicheng@gmail.com>

Description
Calculate the win statistics (win ratio, net benefit and win odds) for prioritized multiple endpoints, plot the win statistics and win proportions over study time if at least one time-to-event endpoint is analyzed, and simulate datasets with dependent endpoints. The package can handle any type of outcomes (continuous, ordinal, binary, time-to-event) and allow users to perform stratified analysis and inverse probability of censoring weighting (IPCW) analysis.

Depends R (>= 3.5.0)

Imports copula,ggplot2,ggpubr,reshape2,survival,stringr,viridis

Suggests knitr, rmarkdown

VignetteBuilder knitr

License GPL (>= 2)

Encoding UTF-8

LazyData true

NeedsCompilation no

Author Ying Cui [aut, cre],
Bo Huang [aut]

Repository CRAN

Date/Publication 2022-07-18 08:00:02 UTC

R topics documented:

`data_binary` *An example with three binary endpoints.*

Description

This is a simulated data example with three binary endpoints.

Usage

```
data("data_binary")
```

Format

A data frame with 250 observations on the following 4 variables.

`id` A vector for the patient id.

`arm` A vector for the treatment groups.

`Y_1` A vector for the outcome of the first endpoint.

`Y_2` A vector for the outcome of the second endpoint.

`Y_3` A vector for the outcome of the third endpoint.

Examples

```
data(data_binary)
str(data_binary)
```

`data_continuous` *An example with three continuous endpoints.*

Description

This is a simulated data example with three continuous endpoints.

Usage

```
data("data_continuous")
```

Format

A data frame with 250 observations on the following 4 variables.

`id` A vector for the patient id.

`arm` A vector for the treatment groups.

`Y_1` A vector for the outcome of the first endpoint.

`Y_2` A vector for the outcome of the second endpoint.

`Y_3` A vector for the outcome of the third endpoint.

Examples

```
data(data_continuous)
str(data_continuous)
```

`data_mix` *An example with a mixture of endpoint types.*

Description

This is a simulated data example with a mixture of two continuous and one time-to-event endpoints. The first endpoint is time-to-event and the second and third endpoints are continuous.

Usage

```
data("data_mix")
```

Format

A data frame with 400 observations on the following 5 variables.

`id` A vector for the patient id.

`arm` A vector for the treatment groups.

`Delta_1` A vector for the event status of the first endpoint (1=event, 0=censored).

`Y_1` A vector for the outcome of the first endpoint.

`Y_2` A vector for the outcome of the second endpoint.

`Y_3` A vector for the outcome of the third endpoint.

Examples

```
data(data_mix)
str(data_mix)
```

`data_mix_stratum` *An example with a mixture of endpoint types with three strata.*

Description

This is a simulated data example with a mixture of two continuous and one time-to-event endpoints with three strata. The first endpoint is time-to-event and the second and third endpoints are continuous.

Usage

```
data("data_mix_stratum")
```

Format

A data frame with 400 observations on the following 6 variables.

`id` A vector for the patient id.

`arm` A vector for the treatment groups.

`stratum` A vector for the stratum.

`Delta_1` A vector for the event status of the first endpoint (1=event, 0=censored).

`Y_1` A vector for the outcome of the first endpoint.

`Y_2` A vector for the outcome of the second endpoint.

`Y_3` A vector for the outcome of the third endpoint.

Examples

```
data(data_mix_stratum)
str(data_mix_stratum)
```

data_tte — *An example with three TTE endpoints.*

Description

This is a simulated data example with three time-to-event endpoints.

Usage

```
data("data_tte")
```

Format

A data frame with 400 observations on the following 7 variables.

- `id` A vector for the patient id.
- `arm` A vector for the treatment groups.
- `Delta_1` A vector for the event status of the first endpoint (1=event, 0=censored).
- `Delta_2` A vector for the event status of the second endpoint (1=event, 0=censored).
- `Delta_3` A vector for the event status of the third endpoint (1=event, 0=censored).
- `Y_1` A vector for the outcome of the first endpoint.
- `Y_2` A vector for the outcome of the second endpoint.
- `Y_3` A vector for the outcome of the third endpoint.

Examples

```
data(data_tte)
str(data_tte)
```

partition_t.plot — *Plot the Win Proportion over the Study Time.*

Description

A plot for the win proportions of the treatment/control group over the study time, for time-to-event endpoints only. The function "win.stat" is called to calculate the win proportions at each time in the plot function.

Usage

```
partition_t.plot(data, Ctime = Inf, arm.name = c(1, 2), priority = c(1, 2),
censoring_adjust = "No", Z_t_trt = NULL, Z_t_con = NULL, tau = 0,
plotTimeUnit = NULL, trt_group = c("both","trt","con"),
win.strategy = NULL, ...)
```

Arguments

`data`
: The analysis dataset which contains the following variables:
  - arm: A vector for the treatment groups. Alternative names for "arm" include "trt", "treat" and "treatment".
  - idA vector for the patient id.
  - stratum: A vector for the stratum.
  - Delta_j: A vector for the event status of the j-th endpoint if the endpoint is a time-to-event outcome (1=event, 0=censored).
  - Y_j: A vector for the outcome of the j-th endpoint, for time-to-event outcome, Y_j is a vector for the observed time.
  - Start_time: A vector for the time when each of the individuals is first accrued to study.

`Ctime`
: A vector of study times, at which the win proportions are to be plotted. Study time for a subject is the time since the subject is accrued, default is Inf.

`arm.name`
: A vector for the two treatments, default to be c(1,2). The first value is for the treatment group, and the second value is for the control group.

`priority`
: Importance order (from the most to the least important). For example, given three endpoints with the importance order as Endpoint 3, Endpoint 2, and Endpoint 1, input priority = c(3,2,1).

`censoring_adjust`
: The method to adjust censoring for the kernal functions. Possible choices are listed below.
  - "No": Without an adjustment.
  - "IPCW": IPCW (inverse-probability-of-censoring weighting) adjustment for censoring using the Kaplan-Meier estimator.
  - "CovIPCW": IPCW adjustment for censoring using the time dependent Cox model.

`Z_t_trt`
: A matrix of the covariate history in the treatment group, each row is a (p+2) vector: the first two columns are id, time, the other p columns are the covariates (vector of length p). The baseline covariates are provided as the observed covariates corresponding to time 0.

`Z_t_con`
: A matrix of the covariate history in the control group, each row is a (p+2) vector: the first two columns are id, time, the other p columns are the covariates (vector of length p). The baseline covariates are provided as the observed covariates corresponding to time 0.

`tau`
: A vector of numerical value for the magnitude of difference to determine win/loss/tie for each endpoint. Tau is applicable for TTE endpoints and continuous endpoints; tau is fixed as 0 for binary endpoints. Default is 0 for all endpoints.

`plotTimeUnit`
: The time units, e.g., "days", "months", "years", default as NULL.

`trt_group`
: An argument to decide the arms for which the win proportions over time are plotted, possible choices include "both", "trt" and "con". Default is "both", i.e., by default the win proportion over time is plotted both for the treatment group and for the control group.

`win.strategy` The strategy to determine the win status. Default as NULL. If NULL, the default win strategy funtion "win.strategy.default" is called, see win.strategy.default for more details. Users can also define their own "win.strategy" function.

`...` Argument passed from user defined functions "win.strategy" if there is any. For instructions on this "win.strategy" function, see win.strategy.default for more details.

Value

A ggplot2 object.

Examples

```
#### An simulated example with two TTE endpoints.
data <- sim.data(n_trt = 200, n_con = 200, n_ep = 2, arm.name = c("A","B"),
ep_type = "tte", cdist.rate = 0.5, sim_method = "copula",
copula_trt=copula::normalCopula(0.9), margins_trt=c("gamma", "beta"),
paramMargins_trt=list(list(shape=2, scale=1),list(shape1=2, shape2=2)),
copula_con=copula::normalCopula(0.9), margins_con=c("gamma", "beta"),
paramMargins_con=list(list(shape=2, scale=1),list(shape1=2, shape2=2)),
max_accrual_time = 5)

partition_t.plot(data, Ctime = seq(0,8,0.2), arm.name = c("A","B"),
priority = c(1,2), tau = 0, plotTimeUnit = "days", trt_group = "both")

#### An simulated example with three TTE endpoints.
data <- sim.data(n_trt = 200, n_con = 200, n_ep = 3, arm.name = c("A","B"),
ep_type = "tte", cdist.rate = 1, sim_method = "copula",
copula_trt=copula::normalCopula(param=c(0.9,0.8,0.95), dim = 3, dispstr = "un"),
margins_trt=c("gamma", "beta", "gamma"),
paramMargins_trt=list(list(shape=2, scale=2),list(shape1=2, shape2=2),list(shape=2, scale=3)),
copula_con=copula::normalCopula(param=c(0.9,0.8,0.95), dim = 3, dispstr = "un"),
margins_con=c("gamma", "beta", "gamma"),
paramMargins_con=list(list(shape=2, scale=1),list(shape1=2, shape2=1),list(shape=2, scale=2)),
max_accrual_time = 5)

partition_t.plot(data, Ctime = c(seq(0,8,0.5),seq(8.1,10,0.1)), arm.name = c("A","B"),
priority = c(3,2,1), tau = 0, plotTimeUnit = "years", trt_group = "trt")
```

`sim.data` *Function for Data Simulation*

Description

A function to simulate a dataset with dependent endpoints. The time-to-event endpoints generated are assumed to have noninformative censoring.

Usage

```
sim.data(randomseed = 12345, n_trt = 200, n_con = 200, n_ep = 2, n_stratum = 1,
arm.name = c(1,2), ep_type, cdist.rate, sim_method = "copula",
copula_trt = NULL, margins_trt = NULL, paramMargins_trt = NULL,
copula_con = NULL, margins_con = NULL, paramMargins_con = NULL,
rate_trt = NULL, rate_con = NULL, max_accrual_time = NULL)
```

Arguments

`randomseed` The random seed.

`n_trt` The number of individuals in the treatment group.

`n_con` The number of individuals in the control group.

`n_ep` The number of endpoints.

`n_stratum` The number of strata. For the simulated dataset, n_stratum is fixed at 1 assuming homogeneous population.

`arm.name` A vector for the labels of the two experimental arms, default to be c(1,2). The first label is for the treatment group, and the second label is for the control group.

`ep_type` A vector for the outcome type for each endpoint. If scalar, the function will treat all the endpoints as the same type. The types of outcome include:

- "tte": Time-to-event outcome, with the default win strategy: the treatment group wins if min(T_trt, C_trt, C_con + tau) > T_con + tau.
- "continuous": Continuous outcome, with the default win strategy: the treatment group wins if Y_trt > Y_con + tau.
- "binary": Binary outcome coded as 0/1, with the default win strategy: 1 is the winner over 0.

`cdist.rate` The censoring time is generated from an exponential distribution. This argument is a vector with the rate of the censoring distribution for each time-to-event endpoint. If scalar, the function will treat all the rate for censoring distribution as the same.

`sim_method` Method used to generate multivariate dependence. Possible choices include "copula" and "tte_exponential"

`copula_trt` an object of "copula" for the treatment group.

`margins_trt` a character vector specifying all the parametric marginal distributions for the treatment group. See details in the R documentation for function "copula::Mvd".

`paramMargins_trt` a list for which each element is a list (or numeric vectors) of named components, giving the parameter values of the marginal distributions for the treatment group. See details in the R documentation for function "copula::Mvd".

`copula_con` Same argument as "copula_trt" for the control group.

`margins_con` Same argument as "margins_trt" for the control group.

`paramMargins_con` Same argument as "paramMargins_trt" for the control group.

rate_trt
: A vector of the rate in the treatment group for each time-to-event endpoint following an exponential distribution when "sim_method" is set to be the option "tte_exponential".

rate_con
: A vector of the rate in the control group for each time-to-event endpoint following an exponential distribution when "sim_method" is set to be the option "tte_exponential".

max_accrual_time
: if specified, simulate the study entry time for each individual from uniform distribution U(0,max_accrual_time).

Details

To learn more about "copula", please refer to a discussion on modelling dependence with copulas with the link https://datascienceplus.com/modelling-dependence-with-copulas/. It shows on a high level how copula works, how to use a copula in R using the copula package and then provides a simple example. Moreover, when "sim_method" is set to be the option "tte_exponential", we simulate two endpoints based on the exponential distribution. Dependence between the two simulated endpoints is introduced, as the earlier endpoint takes the min of the two simulated exponential variables.

Value

data
: The analysis dataset which contains the following variables:

- arm: A vector for the treatment group (trt = 1 | 2), 1 represents the treatment group and 2 represents the control group.
- stratum: A vector for the stratum number. Alternative names for "stratum" include "group", "level" and "grade".
- Delta_j: A vector for the event status of the j-th endpoint if the endpoint is time-to-event outcome (1=event, 0=censored).
- Y_j: A vector for the outcome of the j-th endpoint, for time-to-event outcome, it would be a vector of simulated time.
- Start_time: A vector for the time when each of the individuals is first accrued to study. Valid only if "max_accrual_time" is not NULL.

Examples

```
#### Generate with copula: This example is for three endpoints, noted as Y_1, Y_2, and Y_3,
#### with endpoint type as TTE, TTE and continuous.
#### For both the treatment group and the control group, the correlation coefficients
#### cor(Y_1,Y_2), cor(Y_1,Y_3) and cor(Y_2,Y_3) are 0.9, 0.8 and 0.95, respectively.
#### For each treatment group, the marginal distribution for Y_1, Y_2, and Y_3 are Gamma,
#### Beta and Student t specified as a vector in "margins_trt"/"margins_con". The parameters
#### are specified as a list corresponding to the margianl distributions in "paramMargins_trt"
#### or "paramMargins_con".
sim.data <- sim.data(n_trt = 150, n_con = 100, n_ep = 3, arm.name = c("A","B"),
ep_type = c("tte","tte","continuous"), cdist.rate = 0.5, sim_method = "copula",
copula_trt=copula::normalCopula(param=c(0.9,0.8,0.95), dim = 3, dispstr = "un"),
margins_trt=c("gamma", "beta", "t"),
```

```
paramMargins_trt=list(list(shape=2, scale=1),list(shape1=2, shape2=2),list(df=5)),
copula_con=copula::normalCopula(param=c(0.9,0.8,0.95), dim = 3, dispstr = "un"),
margins_con=c("gamma", "beta", "t"),
paramMargins_con=list(list(shape=1, scale=1),list(shape1=1, shape2=2),list(df=2)),
max_accrual_time = 5)

win_stat <- win.stat(data = sim.data, ep_type = c("tte","tte","continuous"),
arm.name = c("A","B"), priority = c(1,2,3))

#### Generate two TTE endpoints with the more important TTE endpoint expected to occur later
#### with exponential distribution.
sim.data2 <- sim.data(n_trt = 150, n_con = 100, n_ep = 2, arm.name = c("A","B"),
ep_type = c("tte","tte"), cdist.rate = 0.5, sim_method = "tte_exponential",
rate_trt = c(0.2,0.25),rate_con = c(0.4,0.5), max_accrual_time = 5)

win_stat2 <- win.stat(data = sim.data2, ep_type = c("tte","tte"), arm.name = c("A","B"),
priority = c(1,2))
```

`stat_t.plot` *Plot Win Statistics over the Study Time.*

Description

Plot the win statistics as a function of the study time for time-to-event endpoints only. The function "win.stat" is called to calculate the win statistics at each time in the plot function.

Usage

```
stat_t.plot(data, Ctime = Inf, arm.name = c(1,2), priority = c(1,2),
statistic = c("WR","NB","WO"),
Z_t_trt = NULL, Z_t_con = NULL, tau = 0,
weight = c("unstratified","MH-type","wt.stratum1","wt.stratum2","equal"),
censoring_adjust = c("No","IPCW","CovIPCW"),
win.strategy = NULL, plotTimeUnit = NULL,
plot_CI = FALSE, alpha = 0.05, ...)
```

Arguments

`data` The analysis dataset which contains the following variables:

- arm: A vector for the treatment group (trt = 1 | 2), trt is the new treatment. Alternative names for "arm" include "trt", "treat" and "treatment".
- idA vector for the patient id.
- stratum: A vector for the stratum.
- Delta_j: A vector for the event status of the j-th endpoint if the endpoint is a time-to-event outcome (1=event, 0=censored).
- Y_j: A vector for the outcome of the j-th endpoint, for a time-to-event outcome, it would be a vector for time.

- Start_time: A vector for the time when each of the individuals is first accrued to study.

`Ctime` A vector of study times, at which the win proportions are to be plotted. Study time for a subject is the time since the subject is accrued, default as Inf.

`arm.name` A vector for the labels of the two experimental arms, default to be c(1,2). The first label is for the treatment group, and the second label is for the control group.

`priority` Importance order (from the most to the least important). For example, given three endpoints with the importance order as Endpoint 3, Endpoint 2, and Endpoint 1, input priority = c(3,2,1).

`statistic` The win statistic to be plotted.

- "WR": Win ratio.
- "NB": Net benefit.
- "WO": Win odds.

`Z_t_trt` A matrix for the covariate history, each row is a (p+2) vector for one record of each subject in the treatment group. The first two columns are subject id, time, the other columns are the covariates (vector of length p). The baseline covariates are provided as the observed covariates corresponding to time 0.

`Z_t_con` A matrix for the covariate history, each row is a (p+2) vector for one record of each subject in the control group. The first two columns are subject id, time, the other columns are the covariates (vector of length p). The baseline covariates are provided as the observed covariates corresponding to time 0.

`tau` A vector of numerical value for the magnitude of difference to determine win/loss/tie for each endpoint. If tau is input as scalar, the function treat the taus for TTE endpoints and continuous endpoints to be the same and taus for binary endpoints as 0. Default as 0 for all endpoints.

`weight` The weighting method for each stratum. Default is "unstratified" for unstratified analysis. A stratified analysis is performed if other weight option is specified. Other possible choices for this argument are listed below.

- "MH-type": weight the wins with the reciprocal of the stratum size following the Mantel-Haenszel type stratified analysis as described in Dong et al. (2018).
- "wt.stratum1": weight the win statistics with weight equal to the number of subjects in each stratum divided by the total number of subjects.
- "wt.stratum2": weight the win statistics with weight equal to the number of subjects with events (of any TTE endpoint) in each stratum divided by the total number of subjects with events (of any TTE endpoint).
- "equal": set equal weights for all stratum.

`censoring_adjust`

The method to adjust censoring for the kernal functions. Possible choices are listed below.

- "No": Without using the IPCW approach to dealing with the censoring.
- "IPCW": IPCW adjustment for censoring with the Kaplan-Meier estimator.
- "CovIPCW": IPCW adjustment for censoring with the time dependent Cox model.

win.strategy	The strategy to determine the win status. Default as NULL. If NULL, the default win strategy funtion "win.strategy.default" is called, see win.strategy.default for more details. Users can also define their own "win.strategy" function.
plotTimeUnit	The time units, e.g., "days", "months", "years", default as NULL.
plot_CI	If TRUE, plot the pointwise confidence interval, default as FALSE.
alpha	The significance level, default to be 0.05.
...	Argument passed from user defined functions "win.strategy" if there is any. For instructions on this "win.strategy" function, see win.strategy.default for more details.

Value

A ggplot2 object.

Examples

```
#### An simulated example with two TTE endpoints.
data <- sim.data(n_trt = 200, n_con = 200, n_ep = 2, arm.name = c("A","B"),
ep_type = "tte", cdist.rate = 0.5, sim_method = "copula",
copula_trt=copula::normalCopula(0.9), margins_trt=c("gamma", "beta"),
paramMargins_trt=list(list(shape=2, scale=1),list(shape1=2, shape2=2)),
copula_con=copula::normalCopula(0.9), margins_con=c("gamma", "beta"),
paramMargins_con=list(list(shape=2, scale=1),list(shape1=2, shape2=2)),
max_accrual_time = 5)

stat_t.plot(data, arm.name = c("A","B"),priority = c(1,2),
Ctime = seq(2,12,1),plotTimeUnit = "years",statistic = "WR", tau = 0,
weight = "unstratified", censoring_adjust = "No", plot_CI = TRUE)

#### An simulated example with three TTE endpoints.
data <- sim.data(n_trt = 200, n_con = 200, n_ep = 3, arm.name = c("A","B"),
ep_type = "tte", cdist.rate = 0.5, sim_method = "copula",
copula_trt=copula::normalCopula(param=c(0.9,0.8,0.95), dim = 3, dispstr = "un"),
margins_trt=c("gamma", "beta", "t"),
paramMargins_trt=list(list(shape=2, scale=1),list(shape1=2, shape2=2),list(df=5)),
copula_con=copula::normalCopula(param=c(0.9,0.8,0.95), dim = 3, dispstr = "un"),
margins_con=c("gamma", "beta", "t"),
paramMargins_con=list(list(shape=1, scale=1),list(shape1=2, shape2=3),list(df=5)),
max_accrual_time = 5)

stat_t.plot(data, arm.name = c("A","B"),priority = c(3,2,1),
Ctime = seq(1,8,0.5),plotTimeUnit = "years", statistic = "WR",
tau = 0, plot_CI = TRUE)
```

win.stat *Function to Calculate the Win Statistics*

Description

Calculate the win statistics for a mixture type of outcomes including time-to-event outcome, continuous outcome and binary outcome.

Usage

```
win.stat(data, ep_type, Z_t_trt = NULL, Z_t_con = NULL, arm.name = c(1,2),
priority = c(1,2), alpha = 0.05, digit = 5, tau = 0, win.strategy = NULL,
pvalue = c("one-sided","two-sided"),
weight = c("unstratified","MH-type","wt.stratum1","wt.stratum2","equal"),
censoring_adjust = c("No","IPCW","CovIPCW"), var_method = c("Dong et al."),
summary.print = TRUE, ...)
```

Arguments

`data` The analysis dataset which contains the following variables:

- arm: A vector for the treatment groups. Alternative names for "arm" include "trt", "treat" and "treatment".
- idA vector for the patient id.
- stratum: A vector for the stratum.
- Delta_j: A vector for the event status of the j-th endpoint if the endpoint is a time-to-event outcome (1=event, 0=censored).
- Y_j: A vector for the outcome of the j-th endpoint, for time-to-event outcome, Y_j is a vector for the observed time.
- Start_time: A vector for the time when each of the individuals is first accrued to study.

`ep_type` A vector for the outcome type for each endpoint. If scalar, the function will treat all the endpoints as the same type. The types of outcome include:

- "tte": Time-to-event outcome, with the default win strategy: the treatment group wins if min(T_trt, C_trt, C_con + tau) > T_con + tau.
- "continuous": Continuous outcome, with the default win strategy: the treatment group wins if Y_trt > Y_con + tau.
- "binary": Binary outcome coded as 0/1, with the default win strategy: 1 is the winner over 0.

`Z_t_trt` A matrix of the covariate history in the treatment group, each row is a (p+2) vector: the first two columns are subject id, time, the other p columns are the covariates (vector of length p). The baseline covariates are provided as the observed covariates corresponding to time 0.

`Z_t_con` A matrix of the covariate history in the control group, each row is a (p+2) vector: the first two columns are subject id, time, the other p columns are the covariates (vector of length p). The baseline covariates are provided as the observed covariates corresponding to time 0.

- `arm.name` — A vector for the labels of the two experimental arms, default to be c(1,2). The first label is for the treatment group, and the second label is for the control group.
- `priority` — Importance order (from the most to the least important). For example, given three endpoints with the importance order as Endpoint 3, Endpoint 2, and Endpoint 1, input priority = c(3,2,1).
- `alpha` — The significance level, default to be 0.05.
- `digit` — The number of digits for the output, default to be 5.
- `tau` — A vector of numerical value for the magnitude of difference to determine win/loss/tie for each endpoint. Tau is applicable for TTE endpoints and continuous endpoints; tau is fixed as 0 for binary endpoints. Default is 0 for all endpoints.
- `win.strategy` — The strategy to determine the win status. Default as NULL. If NULL, the default win strategy funtion "win.strategy.default" is called, see win.strategy.default for more details. Users can also define their own "win.strategy" function.
- `pvalue` — The p-value type: "one-sided" or "two-sided".
- `weight` — The weighting method for each stratum. Default is "unstratified" for unstratified analysis. A stratified analysis is performed if other weight option is specified. Other possible choices for this argument are listed below.
  - "MH-type": weight the wins with the reciprocal of the stratum size following the Mantel-Haenszel type stratified analysis as described in Dong et al. (2018).
  - "wt.stratum1": weight the win statistics with weight equal to the number of subjects in each stratum divided by the total number of subjects.
  - "wt.stratum2": weight the win statistics with weight equal to the number of subjects with events (of any TTE endpoint) in each stratum divided by the total number of subjects with events (of any TTE endpoint).
  - "equal": set equal weights for all stratum.
- `censoring_adjust` — The method to adjust censoring for the kernal functions. Possible choices are listed below.
  - "No": Without using the IPCW approach to dealing with the censoring.
  - "IPCW": IPCW adjustment for censoring with the Kaplan-Meier estimator.
  - "CovIPCW": IPCW adjustment for censoring with the time dependent Cox model.
- `var_method` — The method to calculate variance.
  - "Dong et al.": The default method based on U-statistics described in Dong et al. (2016)
  - "Luo et al.": This option is available only for data set with two endpoints and without stratum. Obtain the win ratio and net benefits based on Luo et.al (2015)'s method.

  Other methods may be added in future versions.
- `summary.print` — If TRUE, print out a summary of the estimation and inference result for the win statistics; If FALSE, return a list that summarizes the results. Default as TRUE.
- `...` — Argument passed from user defined functions "win.strategy" if there is any. For instructions on this "win.strategy" function, see win.strategy.default for more details.

Details

The arguments of user defined "win.strategy" function must at least include the argument "trt_con" and "priority". "priority" is defined the same as stated in the main function "win.stat". The intermediate analysis dataset "trt_con" for the patient pairs (i.e., unmatched pairs, see Pocock et al., 2012) contains the following variables. Each row represents a pair.

- stratum: A vector for the stratum number of the unmatched pairs.
- pid_trt: A vector for the subject id of the individuals from the treatment group within each unmatched pair.
- pid_con: A vector for the subject id of the individuals from the control group within each unmatched pair.
- Delta_j_trt: A vector for the event status of the j-th endpoint (1=event, 0=censored) for the individuals from the treatment group in each unmatched pair. If the outcome type for the endpoint is continuous/binary, then the event status is 1 for all.
- Delta_j_con: A vector for the event status of the j-th endpoint (1=event, 0=censored) for the individuals from the control group in each unmatched pair. If the outcome type for the endpoint is continuous/binary, then the event status is 1 for all.
- Y_j_trt: A vector for the outcome of the j-th endpoint for the individuals from the treatment group in each unmatched pair. For a time-to-event outcome, it would be a vector of observed time-to-event observations.
- Y_j_con: A vector for the outcome of the j-th endpoint for the individuals from the control group in each unmatched pair. For a time-to-event outcome, it would be a vector of observed time-to-event observations.

Value

`Win_statistic` The win statistics including:

- Win_Ratio: A list for the ratio of the win proportion between the treatment and the control groups.
- Net_Benefit: A list for the difference of the win proportion between the treatment and the control groups.
- Win_Odds: A list for the win odds between the treatment and the control groups.

`pvalue` The p-value for the test statistics including:

- pvalue_WR: p-value for win ratio.
- pvalue_NB: p-value for net benefit.
- pvalue_WO: p-value for win odds.

Note

There may be slight difference between the estimate with Luo et al.'s method and Dong et al.'s method. This is because some pairs may be considered a win for the treatment group with Luo et al.'s method, while considered as a tie with Dong et al.'s method.

References

- Buyse, M., 2010. Generalized pairwise comparisons of prioritized outcomes in the two-sample problem. Statistics in medicine, 29(30), pp.3245-3257.
- 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(2), pp.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(1), pp.139-145.
- Dong, G., Li, D., Ballerstedt, S. and Vandemeulebroecke, M., 2016. A generalized analytic solution to the win ratio to analyze a composite endpoint considering the clinical importance order among components. Pharmaceutical statistics, 15(5), pp.430-437.
- Dong, G., Qiu, J., Wang, D. and Vandemeulebroecke, M., 2018. The stratified win ratio. Journal of biopharmaceutical statistics, 28(4), pp.778-796.
- Dong G, Hoaglin DC, Qiu J, Matsouaka RA, Chang YW, Wang J, Vandemeulebroecke M., 2020. The win ratio: on interpretation and handling of ties. Statistics in iopharmaceutical Research, 12(1), pp.99-106.
- Finkelstein, D.M. and Schoenfeld, D.A., 2019. Graphing the Win Ratio and its components over time. Statistics in medicine, 38(1), pp.53-61.
- Dong, G., Huang, B., Chang, Y.W., Seifu, Y., Song, J. and Hoaglin, D.C., 2020. The win ratio: Impact of censoring and follow-up time and use with nonproportional hazards. Pharmaceutical statistics, 19(3), pp.168-177.
- Dong, G., Mao, L., Huang, B., Gamalo-Siebers, M., Wang, J., Yu, G. and Hoaglin, D.C., 2020. The inverse-probability-of-censoring weighting (IPCW) adjusted win ratio statistic: an unbiased estimator in the presence of independent censoring. Journal of biopharmaceutical statistics, 30(5), pp.882-899.
- Dong, G., Huang, B., Wang, D., Verbeeck, J., Wang, J. and Hoaglin, D.C., 2021. Adjusting win statistics for dependent censoring. Pharmaceutical Statistics, 20(3), pp.440-450.

Examples

```
#### An example with time-to-event outcome.
head(data_tte)
head(Z_t_trt)

### CovIPCW adjustment for dependent censoring
res_tte_covipcw <- win.stat(data = data_tte, ep_type = "tte", arm.name = c("A","B"), tau = 0.1,
Z_t_trt = Z_t_trt, Z_t_con = Z_t_con, priority = c(1:3), alpha = 0.05, digit = 3,
censoring_adjust = "CovIPCW", weight = "unstratified", pvalue = "two-sided")

#### An example with continuous outcome.
head(data_continuous)

res_continuous <- win.stat(data = data_continuous, ep_type = "continuous", arm.name = c("A","B"),
tau = 0, priority = c(1:3), alpha=0.05, digit = 3, weight = "unstratified", pvalue = "two-sided")

#### An example with binary outcome.
```

```
head(data_binary)

res_binary <- win.stat(data = data_binary, ep_type = "binary", arm.name = c("A","B"),
priority = c(1:3), alpha=0.05, digit = 3, weight = "unstratified", pvalue = "two-sided")

#### An example with Luo et al.'s method.
data_luo <- sim.data(n_trt = 200, n_con = 200, n_ep = 2, arm.name = c("A","B"),
ep_type = "tte", cdist.rate = 0.5, sim_method = "copula",
copula_trt=copula::normalCopula(0.9), margins_trt=c("gamma", "beta"),
paramMargins_trt=list(list(shape=2, scale=1),list(shape1=2, shape2=2)),
copula_con=copula::normalCopula(0.9), margins_con=c("gamma", "beta"),
paramMargins_con=list(list(shape=2, scale=1),list(shape1=2, shape2=2)))

res_Luo <- win.stat(data = data_luo, ep_type = "tte", arm.name = c("A","B"), priority = c(1,2),
var_method = "Luo et al.", pvalue = "one-sided")

#### compare the result based on Luo et al.'s method with that based on Dong et al.'s method.
res_Dong <- win.stat(data = data_luo, ep_type = "tte", arm.name = c("A","B"),
priority = c(1,2), pvalue = "one-sided")
```

`win.stat.Luo` *Win Stat Function Provided in the Supplementary Material of Luo et al. (2015).*

Description

A function to calculate the win ratio and the win difference provided in the supplementary material of Luo et al (2015).

Value

`wratio`	The win ratio.
`vwratio`	Variance for the win ratio.
`wdiff`	The win difference.
`vwdiff`	Variance for the win difference.

References

- 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(1), pp.139-145.

`win.strategy.default` *The Default Win Strategy Function.*

Description

An intermediate function to determine the win status for each pair based on the default win strategy. Specifically, one compares each subject in the treatment group with every subject in the control group to determine the win status.

Usage

```
win.strategy.default(trt_con, priority, tau)
```

Arguments

`trt_con` Given N_t subjects in the treatment group and N_c subjects in the control group, there are N_t*N_c pairs, each row represents a pair. The analysis dataset trt_con contains the following variables:

- stratum: A vector for the stratum number of the unmatched pairs.
- pid_trt: A vector for the subject id of the individuals from the treatment group within each unmatched pair.
- pid_con: A vector for the subject id of the individuals from the control group within each unmatched pair.
- Delta_j_trt: A vector for the event status of the j-th endpoint (1=event, 0=censored) for the individuals from the treatment group in each unmatched pair. If the outcome type for the endpoint is continuous/binary, then the event status is 1 for all.
- Delta_j_con: A vector for the event status of the j-th endpoint (1=event, 0=censored) for the individuals from the control group in each unmatched pair. If the outcome type for the endpoint is continuous/binary, then the event status is 1 for all.
- Y_j_trt: A vector for the outcome of the j-th endpoint for the individuals from the treatment group in each unmatched pair. For a time-to-event outcome, it would be a vector of observed time-to-event observations.
- Y_j_con: A vector for the outcome of the j-th endpoint for the individuals from the control group in each unmatched pair. For a time-to-event outcome, it would be a vector of observed time-to-event observations.

`priority` Importance order (from the most to the least important). For example, given three endpoints with the importance order as Endpoint 3, Endpoint 2, and Endpoint 1, input priority = c(3,2,1).

`tau` A vector of numerical value for the magnitude of difference to determine win/loss/tie for each endpoint. Tau is applicable for TTE endpoints and continuous endpoints; tau is fixed as 0 for binary endpoints. Default is 0 for all endpoints.

Details

Default strategy such that the treatment group wins if min(T_trt, C_trt, C_con + tau) > T_con + tau, for time-to-event outcomes; "the larger value wins" for continuous outcome, value 1 wins over the value 0 for binary outcome.

Users can define their own win strategy function and input the function with the argument "win.strategy" in the "win.stat" function.

Value

`win_status` A data frame for the win status of each pair for each endpoint.

`Z_t_con` *Covariate history in the control group.*

Description

This is a simulated data example for the covariate history in the control group.

Usage

```
data("data_tte")
```

Format

A data frame with 796 observations on the following 4 variables.

`id` A vector for the patient id.

`time` A vector for the observed time of the covariate.

`Z1` A vector for the observed value of the first covariate.

`Z2` A vector for the observed value of the second covariate.

Examples

```
data(data_tte)
str(Z_t_con)
```

Z_t_trt *Covariate history in the treatment group.*

Description

This is a simulated data example for the covariate history in the treatment group.

Usage

```
data("data_tte")
```

Format

A data frame with 796 observations on the following 4 variables.

- `id` A vector for the patient id.
- `time` A vector for the observed time of the covariate.
- `Z1` A vector for the observed value of the first covariate.
- `Z2` A vector for the observed value of the second covariate.

Examples

```
data(data_tte)
str(Z_t_trt)
```

Index