Introduction to the R Package `WINS`

Ying Cui and Bo Huang[*]

2022-09-08

A Brief Overview

An inherent limitation of the commonly used composite endpoint approach to analyze prioritized multiple outcomes in clinical trials is the convention to investigate only the event that occurs first. However, the "first-occurred" event is often of less clinical importance. To overcome this limitation, the win statistics (win ratio (Pocock et al., 2012), net benefit (Buyse, 2010) and win odds (Dong et al., 2020)), a class of generalized pairwise comparison methods (Buyse, 2010), were introduced in the literature to analyze prioritized multiple endpoints to determine wins or losses based on the hierarchical order of clinical importance. Specifically, this method compares each subject in the experimental arm with every subject in the control arm by the most important endpoint first, and evaluates lower-priority endpoints if and only if higher-priority endpoints can not determine a win (i.e., a tie) for each pairwise comparison. The win statistics, in particular the win ratio and net benefit, have recently received much attention in methodological research as well as in designs and analyses of Phase III clinical trials. The pioneering work of Finkelstein and Schoenfeld (1999) is closely related to the win statistics.

This document introduces the basic usage of the R package WINS. With the main function win.stat, one can calculate the win statistics and make statistical inference on prioritized multiple endpoints. Although recent methodology research has focused more on time-to-event endpoints, this function can deal with various type of outcomes (time-to-event, continuous, binary, ordinal, or a mixture of them). One can use the functions stat_t.plot, and partition_t.plot to plot the win statistics and win proportions over study time, respectively. A simulation function sim.data can be used to simulate data of multiple endpoints to evaluate the operating characteristics of the methods.

Introduction of the Win Statistics

In this section, we will provide a brief introduction of the win statistics. Denote P_t as the win probability/proportion in the treatment group, and P_c as the win probability/proportion in the control group. The question of interest can then be formulated as testing the null hypothesis $H_0 : P_t = P_c$ versus its alternative. The win statistics are defined based on P_t and P_c as

- Win ratio: $WR = P_t/P_c$.
- Net benefit: $NB = P_t - P_c$.
- Win odds: $WO = \frac{P_t + 0.5(1 - P_t - P_c)}{P_c + 0.5(1 - P_t - P_c)}$.

To obtain the win statistics for a composite of J prioritized outcomes, the counting approach is the most straightforward and intuitive especially when $J > 2$. The basic idea is to compare each subject in the treatment group with every subject in the control group (a total of $N_t N_c$ pairs, where N_t and N_c are the number of subjects in the treatment group and the control group, respectively.). For each pair of subjects, the comparison starts from the most important outcome. The next lower-priority outcome is considered only if the higher-priority outcome results in a tie (i.e., a winner cannot be determined).

First we introduce some necessary notations. Suppose all the J outcomes are time-to-event endpoints. For the treatment group, the observed data are noted as $\boldsymbol{Y} = \{Y_1^{(t)}, \ldots, Y_J^{(t)}\}$ and the event status $\boldsymbol{\delta} = \{\delta_1^{(t)}, \ldots, \delta_J^{(t)}\}$

---

[*]Affiliation: Emory University, Pfizer Inc.

with $\delta_j^{(t)} = 1$ as the event and $\delta_j^{(t)} = 0$ as censoring. Continuous/binary endpoints can be viewed as a special case of the time-to-event outcome with $\delta_j^{(t)} = 1$ for all the subjects. The notations are the same for the control group except that the superscript is $^{(c)}$.

Consider a randomized clinical trial with N_t and N_c subjects in the treatment group and the control group, respectively. We use $i = 1, \ldots, N_t$ for subjects in the treatment group and $l = 1, \ldots, N_c$ for subjects in the Control group. Using the counting approach, we define the kernel functions K and L as follows: $K_{il} = 1$ if subject i in the treatment group wins over subject l in the control group, otherwise $K_{il} = 0$; and $L_{il} = 1$ if subject l wins over subject i, otherwise $L_{il} = 0$. Following the counting approach, we then define the win proportion P_t and P_c as

$$P_t = \frac{\sum_{i=1}^{N_t} \sum_{l=1}^{N_c} K_{il}}{N_t N_c} \text{ and } P_c = \frac{\sum_{i=1}^{N_t} \sum_{l=1}^{N_c} L_{il}}{N_t N_c},$$

where K_{il} and L_{il} are the kernel functions as defined above.

Since the pioneering work by Pocock et.al (2012), the win statistics have received much attention in methodological research. Methods have been developed to correct for bias in the presence of censoring and to extend the use to stratified analysis. For example, Dong et.al (2018) considered a clinical trial with patients randomized into two groups with M strata and proposed the stratified win ratio in a similar way as the Mantel-Haenszel stratified odds ratio. To estimate the win proportions and win statistics in the presence of independent or dependent censoring, Dong et.al (2020, 2021) introduced the inverse-probability-of-censoring weighting (IPCW) adjusted method and CovIPCW-adjusted method using baseline covariates and/or time-dependent covariates that can predict the dependent censoring.

In Pocock et.al (2012), the confidence interval was constructed based on bootstrap resampling, and the hypothesis testing was based on the non-parametric method. To provide a more efficient inference procedure, Luo et al. (2015) developed a close-form variance estimator for the win ratio and the win difference only for cases with two time-to-event endpoints. Later on, Dong et al. (2016) and Bebu and Lachin (2016) developed a more generalized close-form variance estimator for the win statistics which has the flexibility to allow for any number of endpoints.

In the next sections, we will provide some simulated examples to illustrate the use of our R package `WINS` to analyze prioritized multiple outcomes.

Estimate Win Statistics using `win.stat`

In this section, we show how the main function win.stat works. We first provide three examples with three build-in simulated datasets in this R package. The number of endpoints in all these three example datasets are 3. Priority order of multiple endpoints can be specified in the option `priority`. For example, suppose we input endpoint 1, 2 and 3 as `Y_1`, `Y_2` and `Y_3`, respectively. `priority=c(2,1,3)` specifies that `Y_2` is the most important and `Y_3` is the least important.

An example with three binary endpoints

We consider the values 1 vs 0 as the default setting for binary variables since oftentimes the value 1 represents response to the drug, and the default win strategy is that the value 1 wins over the value 0.

The example dataset `data_binary` contains 3 binary outcomes.

```
head(data_binary)
#>   id arm Y_1 Y_2 Y_3
#> 1  1   A   0   0   0
#> 2  2   A   0   0   1
#> 3  3   A   1   1   0
#> 4  4   A   1   1   0
#> 5  5   A   1   1   1
#> 6  6   A   0   0   0
```

```
res_binary <- win.stat(data = data_binary, ep_type = "binary", priority = c(1:3),
                       weight = "unstratified", arm.name = c("A","B"), alpha=0.05,
                       digit = 3,pvalue = "two-sided")
#> The outcome type for all the endpoints:  binary
#> This analysis is unstratified.
#>
#>  Win Ratio : 2.299
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win ratio:  1.511
#>  Upper limit of 95 % CI of the win ratio:  3.497
#>
#>
#>  Net Benefit : 0.278
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the net benefit:  0.130
#>  Upper limit of 95 % CI of the net benefit:  0.426
#>
#>
#>  Win Odds : 1.769
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win odds:  1.316
#>  Upper limit of 95 % CI of the win odds:  2.379
#>
```

An example with three continuous endpoints

The default win strategy for continuous outcomes is "the larger value wins".

The example code is presented below.

```
head(data_continuous)
#>   id arm       Y_1        Y_2          Y_3
#> 1  1   A 0.4621177 0.2069386 -0.52627613
#> 2  2   A 0.3258938 0.2595162 -0.11247766
#> 3  3   A 2.4887816 0.5406375 -0.34632514
#> 4  4   A 3.9419121 0.4956837 -0.44683113
#> 5  5   A 2.3304126 0.5481473 -0.02682927
#> 6  6   A 0.3168587 0.1186624 -1.85656323

res_continuous <- win.stat(data = data_continuous, ep_type = "continuous",
                           arm.name = c("A","B"),tau = 0, priority = c(1:3),
                           weight = "unstratified", alpha=0.05, digit = 3,
                           pvalue = "two-sided")
#> The outcome type for all the endpoints:  continuous
#> This analysis is unstratified.
#>
#>  Win Ratio : 1.695
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win ratio:  1.241
#>  Upper limit of 95 % CI of the win ratio:  2.316
```

```
#>
#>
#>   Net Benefit : 0.258
#>
#>   two-sided p-value is:   0.001
#>   Lower limit of 95 % CI of the net benefit:   0.102
#>   Upper limit of 95 % CI of the net benefit:   0.414
#>
#>
#>   Win Odds : 1.695
#>
#>   two-sided p-value is:   < 0.001
#>   Lower limit of 95 % CI of the win odds:   1.241
#>   Upper limit of 95 % CI of the win odds:   2.316
#>
```

Note that in this example, the $WR = WO$ because there are no ties for continuous outcomes.

An example with three time-to-event endpoints

For time-to-event (TTE) outcomes, the adjustment for censoring can be specified in the option `censoring_adjust`.

For illustration, we first calculate the win statistics without using the IPCW approach to dealing with the censoring. The example code is presented below.

```
head(data_tte)
#>   id arm Delta_1 Delta_2 Delta_3       Y_1       Y_2       Y_3
#> 1  1   A       0       0       0 0.1268054 2.9063267 0.2605294
#> 2  2   A       0       1       1 1.0137019 1.7945845 1.5220050
#> 3  3   A       1       0       0 1.4323415 1.5009229 0.6054241
#> 4  4   A       1       1       0 1.6961593 5.1323268 1.9331923
#> 5  5   A       1       0       1 3.5786849 0.4329275 0.6391820
#> 6  6   A       1       1       0 0.5615477 8.2259638 0.5401791


### Without using the IPCW approach to dealing with the censoring
res_tte <- win.stat(data = data_tte, ep_type = "tte", arm.name = c("A","B"),
                    tau = 0.1, priority = c(1:3), alpha = 0.05, digit = 3,
                    weight = "unstratified", censoring_adjust = "No",
                    pvalue = "two-sided")
#> The outcome type for all the endpoints:   tte
#> This analysis is unstratified.
#>
#>   Win Ratio : 1.902
#>
#>   two-sided p-value is:   < 0.001
#>   Lower limit of 95 % CI of the win ratio:   1.498
#>   Upper limit of 95 % CI of the win ratio:   2.414
#>
#>
#>   Net Benefit : 0.279
#>
#>   two-sided p-value is:   < 0.001
#>   Lower limit of 95 % CI of the net benefit:   0.172
#>   Upper limit of 95 % CI of the net benefit:   0.386
```

```
#>
#>
#>  Win Odds : 1.775
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win odds:  1.432
#>  Upper limit of 95 % CI of the win odds:  2.198
#>
```

When censoring is assumed to be independent of the outcomes, we provide the option `censoring_adjust =`
`"IPCW"` to calculate IPCW-adjusted win statistics (Dong et al., 2020). The example code is presented below.

```
### IPCW adjustment for independent censoring
res_tte_ipcw <- win.stat(data = data_tte, ep_type = "tte", arm.name = c("A","B"),
                         tau = 0.1, priority = c(1:3), alpha = 0.05, digit = 3,
                         weight = "unstratified", censoring_adjust = "IPCW",
                         pvalue = "two-sided")
#> The outcome type for all the endpoints:  tte
#> This analysis is unstratified.
#>
#>  Win Ratio : 1.874
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win ratio:  1.491
#>  Upper limit of 95 % CI of the win ratio:  2.356
#>
#>
#>  Net Benefit : 0.304
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the net benefit:  0.190
#>  Upper limit of 95 % CI of the net benefit:  0.419
#>
#>
#>  Win Odds : 1.874
#>
#>  two-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win odds:  1.491
#>  Upper limit of 95 % CI of the win odds:  2.356
#>
```

When censoring is suspected to be dependent on baseline or time dependent covariates which correlate with
the outcomes, we provide the option `censoring_adjust = "CovIPCW"` to calculate CovIPCW-adjusted win
statistics (Dong et al., 2021). The example code is presented below.

```
head(Z_t_trt)
#>   id time        Z1         Z2
#> 1  1  0.0 0.9257546 0.74457231
#> 2  2  0.0 0.1970530 0.56999281
#> 3  2  0.2 0.7914286 0.65883216
#> 4  2  0.4 0.2435229 0.01919255
#> 5  2  0.6 0.3393664 0.69654040
#> 6  2  0.8 0.1932459 0.98218826


### CovIPCW adjustment for dependent censoring
```

```
res_tte_covipcw <- win.stat(data = data_tte, ep_type = "tte", tau = 0.1,
                            arm.name = c("A","B"), weight = "unstratified",
                            Z_t_trt = Z_t_trt, Z_t_con = Z_t_con,
                            priority = c(1:3), alpha = 0.05, digit = 3,
                            censoring_adjust = "CovIPCW", pvalue = "two-sided")
#> The outcome type for all the endpoints:  tte
#> This analysis is unstratified.
#>
#>   Win Ratio : 1.845
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win ratio:  1.476
#>   Upper limit of 95 % CI of the win ratio:  2.308
#>
#>
#>   Net Benefit : 0.297
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the net benefit:  0.185
#>   Upper limit of 95 % CI of the net benefit:  0.409
#>
#>
#>   Win Odds : 1.845
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win odds:  1.476
#>   Upper limit of 95 % CI of the win odds:  2.308
#>
```

In the last two examples, there is $WR = WO$ because we adjusted the win proportions so that the tie proportion is not negative (i.e. equal to zero after adjustment) after IPCW adjustment.

These three examples presented above show the basic usage of the R function `win.stat` with various types of endpoints. In the next example, we provide a more complex data example.

An example with a mixture of endpoint types: two continuous and one TTE

This example is for a simulated dataset with three endpoints as a mixture of two continuous endpoints and one TTE endpoint.

The example code is presented below.

```
head(data_mix)
#>   id arm Delta_1       Y_1       Y_2        Y_3
#> 1  1   A       0 0.7252257 0.5557491  0.3580795
#> 2  2   A       1 0.4242003 0.1810918 -1.4360364
#> 3  3   A       1 0.6900970 0.2399882 -1.8876155
#> 4  4   A       0 2.8391707 0.7336988  0.7722108
#> 5  5   A       0 0.6134876 0.3533288 -0.2674180
#> 6  6   A       1 0.4434930 0.2379743 -1.2457559

res_mix <- win.stat(data = data_mix, ep_type = c("tte","continuous","continuous"),
                    arm.name = c("A","B"), tau = 0.1, priority = c(1:3),
                    alpha = 0.05, digit = 3, censoring_adjust = "No",
                    weight = "unstratified", pvalue = "two-sided")
#> This analysis is unstratified.
```

```
#>
#>   Win Ratio : 1.936
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win ratio:   1.517
#>   Upper limit of 95 % CI of the win ratio:   2.470
#>
#>
#>   Net Benefit : 0.319
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the net benefit:   0.197
#>   Upper limit of 95 % CI of the net benefit:   0.441
#>
#>
#>   Win Odds : 1.936
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win odds:   1.517
#>   Upper limit of 95 % CI of the win odds:   2.470
#>
```

When there is at least one TTE endpoints among all, the option censoring_adjust to calculate IPCW-adjusted and CovIPCW-adjusted win statistic is available.

The example code is presented below.

```
### IPCW adjustment for independent censoring
res_mix_ipcw <- win.stat(data = data_mix,
                         ep_type = c("tte","continuous","continuous"),
                         arm.name = c("A","B"), tau = 0.1, priority = c(1:3),
                         alpha = 0.05, digit = 3, censoring_adjust = "IPCW",
                         weight="unstratified", pvalue = "two-sided")
#> This analysis is unstratified.
#>
#>   Win Ratio : 1.936
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win ratio:   1.517
#>   Upper limit of 95 % CI of the win ratio:   2.470
#>
#>
#>   Net Benefit : 0.319
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the net benefit:   0.197
#>   Upper limit of 95 % CI of the net benefit:   0.441
#>
#>
#>   Win Odds : 1.936
#>
#>   two-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win odds:   1.517
#>   Upper limit of 95 % CI of the win odds:   2.470
#>
```

An example with a mixture of endpoint types: two continuous and one TTE with three strata

In this example, besides a mixture of endpoint types, there are three strata in the data. By default, the stratified win statistics proposed in Dong et al. (2018) are calculated.

Let $N_t^{(m)}$ and $N_c^{(m)}$ denote the number of subjects in the treatment group and the control group of the mth stratum, respectively, and let $N^{(m)} = N_t^{(m)} + N_c^{(m)}$ denote the total number of subjects in the mth stratum for $m = 1, \ldots, M$. Within each stratum, define the kernal functions $K_{il}^{(m)}(\boldsymbol{Y}_i^{(m)}, \boldsymbol{\delta}_i^{(m)}, \boldsymbol{Y}_l^{(m)}, \boldsymbol{\delta}_l^{(m)})$ and $L_{il}^{(m)}(\boldsymbol{Y}_i^{(m)}, \boldsymbol{\delta}_i^{(m)}, \boldsymbol{Y}_l^{(m)}, \boldsymbol{\delta}_l^{(m)})$ in a similar way as the case without stratification. The stratified win ratio is then defined as

$$WR_{MH} = \frac{\sum_{m=1}^{M} w^{(m)} n_t^{(m)}}{\sum_{m=1}^{M} w^{(m)} n_c^{(m)}},$$

where $n_t^{(m)} = \sum_{i=1}^{N_t^{(m)}} \sum_{l=1}^{N_c^{(m)}} K_{il}^{(m)}$, $n_c^{(m)} = \sum_{i=1}^{N_t^{(m)}} \sum_{l=1}^{N_c^{(m)}} L_{il}^{(m)}$, and $\{w^{(m)}, m = 1, \ldots, M\}$ represent the weight assigned to the M strata.

Firstly, we provide the option to assign equal weights for all the strata, e.g., $w(m) = \frac{1}{M}$. The example code is presented below.

```
res_mix_equal <- win.stat(data = data_mix_stratum,
                          ep_type = c("tte","continuous","continuous"),
                          arm.name = c("A","B"), tau = 0.1, priority = c(1:3),
                          alpha = 0.05, digit = 3, censoring_adjust = "No",
                          weight = "equal", pvalue = "one-sided")
#>
#>  Win Ratio : 1.951
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win ratio:  1.528
#>  Upper limit of 95 % CI of the win ratio:  2.491
#>
#>
#>  Net Benefit : 0.322
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the net benefit:  0.200
#>  Upper limit of 95 % CI of the net benefit:  0.444
#>
#>
#>  Win Odds : 1.951
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win odds:  1.528
#>  Upper limit of 95 % CI of the win odds:  2.491
#>
```

Then we consider the option to assign the weight for the m-th stratum as $w^{(m)} = 1/N^{(m)}$. The example code is presented below.

```
head(data_mix_stratum)
#>   id arm stratum Delta_1       Y_1       Y_2        Y_3
#> 1  1   A       2       0 0.7252257 0.5557491  0.3580795
#> 2  2   A       3       1 0.4242003 0.1810918 -1.4360364
#> 3  3   A       2       1 0.6900970 0.2399882 -1.8876155
#> 4  4   A       2       0 2.8391707 0.7336988  0.7722108
#> 5  5   A       1       0 0.6134876 0.3533288 -0.2674180
```

```
#> 6  6    A      3      1 0.4434930 0.2379743 -1.2457559

res_mix_MH <- win.stat(data = data_mix_stratum,
                       ep_type = c("tte","continuous","continuous"),
                       arm.name = c("A","B"), tau = 0.1, priority = c(1:3),
                       alpha = 0.05, digit = 3, censoring_adjust = "No",
                       weight = "MH-type", pvalue = "one-sided")
#>
#>   Win Ratio : 1.952
#>
#>   one-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win ratio:   1.529
#>   Upper limit of 95 % CI of the win ratio:   2.492
#>
#>
#>   Net Benefit : 0.323
#>
#>   one-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the net benefit:   0.200
#>   Upper limit of 95 % CI of the net benefit:   0.445
#>
#>
#>   Win Odds : 1.952
#>
#>   one-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win odds:   1.529
#>   Upper limit of 95 % CI of the win odds:   2.492
#>
```

In the above example, the stratified win ratio is proposed in a similar way as the Mantel-Haenszel stratified odds ratio. Specifically, given M stratum, the stratum-specific wins are weighted to estimate the stratified win ratio.

We then provide another option to define the stratified win ratio, i.e., the weight is applied directly to the stratum-specific win statistics with the number of subjects. As an example, the stratified win ratio is defined as

$$WR_{wt,1} = \sum_{m=1}^{M} w_1^{(m)} WR^{(m)} = \sum_{m=1}^{M} \frac{w_1^{(m)} n_t^{(m)}}{n_c^{(m)}}$$

with $w_1^{(m)} = \frac{N^{(m)}}{\sum_{m=1}^{M} N^{(m)}}$.

The example code is presented below.

```
res_mix_wt1 <- win.stat(data = data_mix_stratum,
                        ep_type = c("tte","continuous","continuous"),
                        arm.name = c("A","B"), tau = 0.1, priority = c(1:3),
                        alpha = 0.05, digit = 3, censoring_adjust = "No",
                        weight = "wt.stratum1", pvalue = "one-sided")
#>
#>   Win Ratio : 1.961
#>
#>   one-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win ratio:   1.731
#>   Upper limit of 95 % CI of the win ratio:   2.221
#>
```

```
#>
#>  Net Benefit : 0.322
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the net benefit:  0.200
#>  Upper limit of 95 % CI of the net benefit:  0.445
#>
#>
#>  Win Odds : 1.961
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win odds:  1.731
#>  Upper limit of 95 % CI of the win odds:  2.221
#>
```

When there exist at least one TTE endpoints, we further provide a third option to set weight directly to the win statistics in each stratum relative to the number of events, i.e., the stratified win ratio is defined as

$$
WR_{wt,2} = \sum_{m=1}^{M} w_2^{(m)} WR^{(m)} = \sum_{m=1}^{M} \frac{w_2^{(m)} n_t^{(m)}}{n_c^{(m)}}
$$

with $w_2^{(m)} = \frac{N_{event}^{(m)}}{\sum_{m=1}^{M} N_{event}^{(m)}}$.

The example code is presented below.

```r
res_mix_wt2 <- win.stat(data = data_mix_stratum,
                        ep_type = c("tte","continuous","continuous"),
                        arm.name = c("A","B"), tau = 0.1, priority = c(1:3),
                        alpha = 0.05, digit = 3, censoring_adjust = "No",
                        weight = "wt.stratum2", pvalue = "one-sided")
#>
#>  Win Ratio : 1.967
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win ratio:  1.736
#>  Upper limit of 95 % CI of the win ratio:  2.228
#>
#>
#>  Net Benefit : 0.324
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the net benefit:  0.201
#>  Upper limit of 95 % CI of the net benefit:  0.446
#>
#>
#>  Win Odds : 1.967
#>
#>  one-sided p-value is:  < 0.001
#>  Lower limit of 95 % CI of the win odds:  1.736
#>  Upper limit of 95 % CI of the win odds:  2.228
#>
```

Data simulation with the function `sim.data`

In this section, we provide two examples to show how to use `sim.data` to simulate data. We provide two options. One is based on exponential distributions, the other simulates correlated endpoints using the existing R package `copula`. The accrual time is assumed to follow a certain distribution, for example, uniform distribution.

The first option focuses on generating two TTE endpoints Y_1 and Y_2 with the more important TTE endpoint Y_2 expected to occur later. Also, the two endpoints are in a semi-competing risk setting with Y_2 as the terminal endpoint (i.e., Y_1 cannot occur later than Y_2). A real example for those two endpoints is the composite of progression (Y_1) and death (Y_2) in oncology clinical trials. If death is the event that happened first, then Y_1 is censored by death; while the reverse is not true.

The example code is presented below.

```
#### Generate two TTE endpoints with the more important TTE endpoint expected to occur
#### later with exponential distributions.
sim.data_tte <- 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(0.3,0.2), rate_con = c(0.5,0.4),
                         max_accrual_time = 5)

win_stat_sim_tte <- win.stat(data = sim.data_tte, ep_type = c("tte","tte"),
                             arm.name = c("A","B"), digit = 3, priority = c(2,1))
#> This analysis is unstratified.
#>
#>   Win Ratio : 1.893
#>
#>   one-sided p-value is:  0.001
#>   Lower limit of 95 % CI of the win ratio:  1.251
#>   Upper limit of 95 % CI of the win ratio:  2.866
#>
#>
#>   Net Benefit : 0.161
#>
#>   one-sided p-value is:  0.002
#>   Lower limit of 95 % CI of the net benefit:  0.053
#>   Upper limit of 95 % CI of the net benefit:  0.269
#>
#>
#>   Win Odds : 1.383
#>
#>   one-sided p-value is:  0.002
#>   Lower limit of 95 % CI of the win odds:  1.115
#>   Upper limit of 95 % CI of the win odds:  1.717
#>
```

The second option is to generate the data with the R package `copula`. This option works for any number of endpoints and is able to simulate the correlation between the endpoints, however there will not be competing risk among the simulated endpoints.

The below is an example with 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 marginal distributions in "paramMargins_trt" or

"paramMargins_con".

The example code is presented below.

```
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"), digit = 3, priority = c(1,2,3))
#> This analysis is unstratified.
#>
#>   Win Ratio : 2.333
#>
#>   one-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win ratio:  1.671
#>   Upper limit of 95 % CI of the win ratio:  3.256
#>
#>
#>   Net Benefit : 0.400
#>
#>   one-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the net benefit:  0.233
#>   Upper limit of 95 % CI of the net benefit:  0.567
#>
#>
#>   Win Odds : 2.333
#>
#>   one-sided p-value is:  < 0.001
#>   Lower limit of 95 % CI of the win odds:  1.671
#>   Upper limit of 95 % CI of the win odds:  3.256
#>
```

Plot the Win Statistics/Win Proportions over Time

Win statistics and win proportions are dependent on follow-up time (e.g., Oakes, 2016), and they can be graphically presented (Finkelstein and Schoenfeld, 2019). In this section, we show the usage of the two plot functions `stat_t.plot` and `partition_t.plot` for TTE endpoints only. The two functions are able to plot the win statistics of stated with its point-wise 95% confidence interval, as well as the win proportion in each treatment group over study time. If the study entry time for each individual is not provided in the dataset, then by default all the individuals are assumed to enter the study at the same time.

The example code is presented below.

```
#### 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)
#> [1] "The outcome type for all the endpoints: tte"

stat_t.plot(data, arm.name = c("A","B"),priority = c(3,2,1),
            Ctime = seq(2,8,0.2),plotTimeUnit = "years",
            statistic = "WR", tau = 0, plot_CI = TRUE)
#> This analysis is unstratified.
```
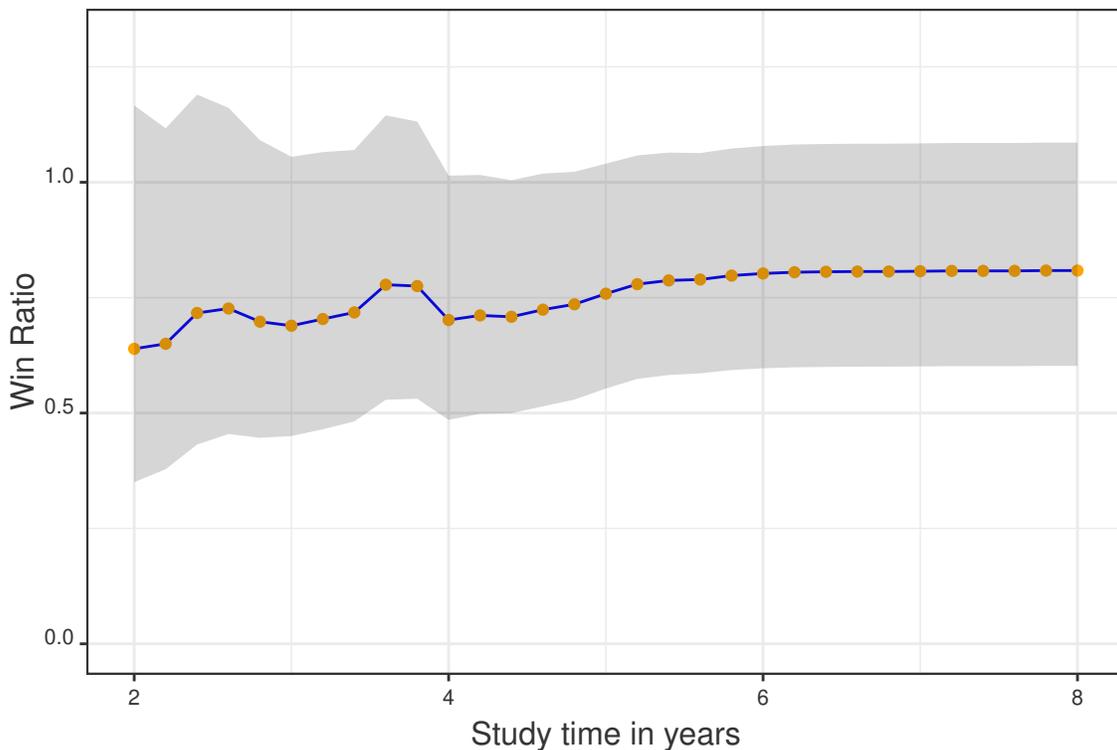


```
#> $statistic
#> [1] "WR"
#>
#> $values
#>    time  win_stat  lower_ci upper_ci
#> 1   2.0 0.6391489 0.3501887 1.166546
#> 2   2.2 0.6500593 0.3783346 1.116940
```

13

```
#> 3    2.4 0.7167798 0.4317604 1.189950
#> 4    2.6 0.7264924 0.4544961 1.161267
#> 5    2.8 0.6979066 0.4463499 1.091237
#> 6    3.0 0.6891560 0.4500785 1.055229
#> 7    3.2 0.7037225 0.4648179 1.065418
#> 8    3.4 0.7179833 0.4817856 1.069978
#> 9    3.6 0.7780083 0.5287173 1.144840
#> 10   3.8 0.7751268 0.5309108 1.131681
#> 11   4.0 0.7015848 0.4852616 1.014342
#> 12   4.2 0.7115139 0.4983674 1.015821
#> 13   4.4 0.7085259 0.4997771 1.004466
#> 14   4.6 0.7240062 0.5144887 1.018847
#> 15   4.8 0.7356322 0.5291556 1.022676
#> 16   5.0 0.7585466 0.5530718 1.040358
#> 17   5.2 0.7792752 0.5738776 1.058187
#> 18   5.4 0.7872996 0.5823905 1.064304
#> 19   5.6 0.7893421 0.5859536 1.063328
#> 20   5.8 0.7978184 0.5931881 1.073039
#> 21   6.0 0.8024024 0.5969874 1.078498
#> 22   6.2 0.8050622 0.5990741 1.081878
#> 23   6.4 0.8061084 0.5999812 1.083052
#> 24   6.6 0.8065872 0.6004244 1.083538
#> 25   6.8 0.8067061 0.6005489 1.083633
#> 26   7.0 0.8072372 0.6010138 1.084221
#> 27   7.2 0.8079595 0.6016086 1.085088
#> 28   7.4 0.8079595 0.6016086 1.085088
#> 29   7.6 0.8079595 0.6016086 1.085088
#> 30   7.8 0.8086986 0.6022261 1.085960
#> 31   8.0 0.8086986 0.6022261 1.085960
```

```
partition_t.plot(data, Ctime = c(seq(0,9,0.5),seq(9.1,11,0.1)), arm.name = c("A","B"),
priority = c(3,2,1), tau = 0, plotTimeUnit = "years", trt_group = "trt")
```

The win proportion for the treatment group
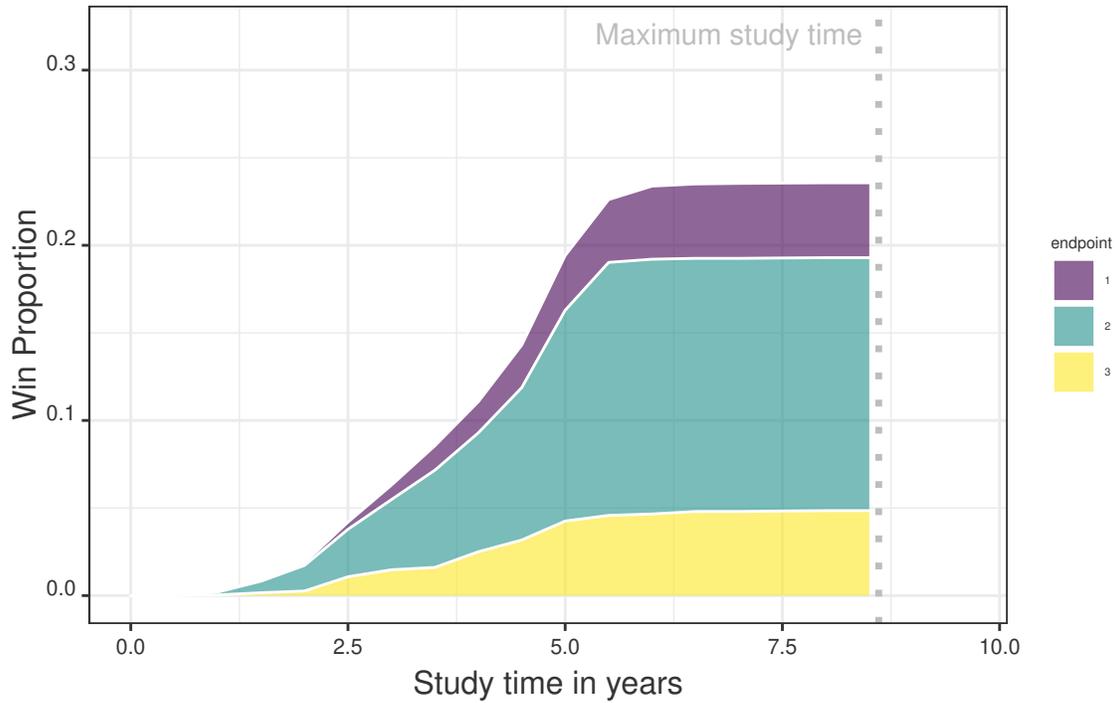

```
#> $win_trt_t
#>     time endpoint1 endpoint2 endpoint3
#> 1   0.0  0.000000  0.000000  0.000000
#> 2   0.5  0.000000  0.000000  0.000000
#> 3   1.0  0.000025  0.001625  0.000375
#> 4   1.5  0.000325  0.006625  0.001625
#> 5   2.0  0.001525  0.014625  0.002625
#> 6   2.5  0.004100  0.027125  0.010750
#> 7   3.0  0.008250  0.040075  0.014750
#> 8   3.5  0.013950  0.055500  0.016100
#> 9   4.0  0.017775  0.067875  0.025025
#> 10  4.5  0.024300  0.087000  0.031725
#> 11  5.0  0.031225  0.120375  0.042550
#> 12  5.5  0.035875  0.144450  0.045775
#> 13  6.0  0.041750  0.145450  0.046600
#> 14  6.5  0.042475  0.144550  0.048000
#> 15  7.0  0.042800  0.144525  0.048025
#> 16  7.5  0.042750  0.144450  0.048300
#> 17  8.0  0.042750  0.144425  0.048500
#> 18  8.5  0.042750  0.144425  0.048500
#>
#> $win_con_t
#>     time endpoint1 endpoint2 endpoint3
#> 1   0.0  0.000000  0.000000  0.000000
#> 2   0.5  0.000000  0.000125  0.000000
#> 3   1.0  0.000000  0.004925  0.000000
#> 4   1.5  0.000000  0.017650  0.000000
#> 5   2.0  0.000000  0.029375  0.000000
#> 6   2.5  0.000575  0.057075  0.000150
#> 7   3.0  0.000925  0.090300  0.000300
```

```
#> 8    3.5  0.001375  0.113375  0.000375
#> 9    4.0  0.001975  0.155200  0.000575
#> 10   4.5  0.002525  0.200900  0.000675
#> 11   5.0  0.002925  0.250850  0.002175
#> 12   5.5  0.005475  0.280350  0.003225
#> 13   6.0  0.005800  0.281950  0.003625
#> 14   6.5  0.005850  0.281650  0.003950
#> 15   7.0  0.005900  0.281475  0.004175
#> 16   7.5  0.005900  0.281350  0.004225
#> 17   8.0  0.005900  0.281300  0.004225
#> 18   8.5  0.005900  0.281300  0.004225
#>
#> $win_tie_t
#>      time proportion of ties
#> 1    0.0            1.000000
#> 2    0.5            0.999875
#> 3    1.0            0.993050
#> 4    1.5            0.973775
#> 5    2.0            0.951850
#> 6    2.5            0.900225
#> 7    3.0            0.845400
#> 8    3.5            0.799325
#> 9    4.0            0.731575
#> 10   4.5            0.652875
#> 11   5.0            0.549900
#> 12   5.5            0.484850
#> 13   6.0            0.474825
#> 14   6.5            0.473525
#> 15   7.0            0.473100
#> 16   7.5            0.473025
#> 17   8.0            0.472900
#> 18   8.5            0.472900
#>
#> $max_study_time
#> [1] 8.608721
```

Acknowledgement

Reference

- Finkelstein, D.M. and Schoenfeld, D.A., 1999. Combining mortality and longitudinal measures in clinical trials. Statistics in medicine, 18(11), pp.1341-1354.

- 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.

- 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(1), pp.178-187.

- 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.

- 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., Hoaglin, DC., Qiu, J., Matsouaka, RA., Chang, Y., Wang, J., Vandemeulebroecke, M., 2020. The win ratio: on interpretation and handling of ties. Statistics in Biopharmaceutical Research, 12(1), pp.99-106

- 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.