

Package ‘maxcombo’

May 8, 2026

Type Package

Title The Group Sequential Max-Combo Test for Comparing Survival Curves

Version 1.0

Date 2020-12-07

Author Tom Prior

Maintainer Tom Prior <tomjamesprior@gmail.com>

Description Functions for comparing survival curves using the max-combo test at a single time-point or repeatedly at successive respective timepoints while controlling type I error (i.e., the group sequential setting), as published by Prior (2020) <doi:10.1177/0962280220931560>. The max-combo test is a generalization of the weighted log-rank test, which itself is a generalization of the log-rank test, which is a commonly used statistical test for comparing survival curves, e.g., during or after a clinical trial as part of an effort to determine if a new drug or therapy is more effective at delaying undesirable outcomes than an established drug or therapy or a placebo.

License GPL-2

Imports magrittr, purrr, rlang, mstate, MCMCpack, dplyr, mvtnorm, survival

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2020-12-09 10:10:06 UTC

Contents

maxcombo-package	2
oogetdoublemaxcombocutoff	3
oogetdoublemaxcombocutoffgroupsequential	15
oogetdoublemaxcombobotestpvalue	34
oogetdoublemaxcomboboteststatistic	44

Index	53
--------------	-----------

maxcombo-package

The Group Sequential Max-Combo Test for Comparing Survival Curves

Description

This package contains functions for comparing survival curves using the max-combo test at a single timepoint or repeatedly at successive respective timepoints while controlling type I error (i.e., the group sequential setting), as published by Tom Prior (Prior 2020 Group sequential monitoring based on the maximum of weighted log-rank statistics with the Fleming-Harrington class of weights in oncology clinical trials). The max-combo test is a generalization of the weighted log-rank test, which itself is a generalization of the log-rank test, which is a commonly used statistical test for comparing survival curves, e.g., during or after a clinical trial as part of an effort to determine if a new drug or therapy is more effective at delaying undesirable outcomes than an established drug or therapy or a placebo.

Details

This package contains the following four functions:

- [oogetdoublemaxcombostatistic](#)
- [oogetdoublemaxcombopvalue](#)
- [oogetdoublemaxcombocutoff](#)
- [oogetdoublemaxcombocutoffgroupsequential](#)

A reasonable way to get started with using this package is to read the documentation for those functions in the order they are listed above.

Author(s)

Tom Prior <tprior1@its.jnj.com or tomjamesprior@gmail.com>

Maintainer: Tom Prior <tprior1@its.jnj.com or tomjamesprior@gmail.com>

References

Prior 2020 Group sequential monitoring based on the maximum of weighted log-rank statistics with the Fleming-Harrington class of weights in oncology clinical trials.

(Available at: <https://doi.org/10.1177/0962280220931560>)

`oogetdoublemaxcombocutoff`*Compute the cutoff value for the max-combo test statistic at a single timepoint*

Description

This function computes the cutoff value for a user-specified nominal type I error rate (also known as a user-specified "alpha") for the max-combo test statistic for comparing times to an event from two different arms at a single timepoint; please note that this is *not* a cutoff for a group sequential test procedure involving two or more timepoints, but rather a cutoff for a scenario in which there is only one planned analysis timepoint. If the true survival curves from the two arms are not at all different, then the probability, before performing the experiment, that the max-combo statistic will be greater than the cutoff value returned by this function, is less than or approximately equal to (but typically approximately equal to) the user-specified nominal type I error rate (e.g., 0.025 for a confirmatory clinical trial). The max-combo test statistic is the maximum of multiple standardized weighted log-rank test statistics. The user of the function is expected to provide the function with a data frame containing the time-to-event data for the two arms; additionally, this data frame is expected to be in a particular format (see more details below). Additionally, the user of the function is expected to provide a list of one or more weighting functions to use for the respective one or more weighted log-rank tests; in particular this function only allows the user to specify a weighting function that takes as input the Kaplan-Meier estimate for the survival curve obtained by pooling both of the two arms (a double that is between 0 and 1, inclusive) and provides as output a weight (a double that is typically greater than or equal to zero); note that the Fleming-Harrington class of weighting functions falls within what can be specified in this way. If no list of weighting functions is provided, a single weighting function that is identically 1 is used, which means that the cutoff that is returned is exactly equal to the cutoff for the usual standardized log-rank test statistic (e.g., approximately 1.96 for a user-specified nominal type I error rate of 0.025).

The supplied data frame is expected to have one row for each subject. The data frame should have the following columns:

- `id` - a variable that takes a unique value for each subject
- `treated` - should be 0 for subjects in the control arm and 1 for subjects in the experimental arm.
- `Atime` - the absolute time the subject enters the study
- `Btime` - the absolute time of an event or censoring, whichever comes first
- `Bobserved` - should be TRUE if the event was observed
- `Ctime` - the absolute time of an event or censoring, whichever comes first
- `Cobserved` - should be TRUE if censoring occurred (and so the event was not observed)

The columns in the data frame do not have to be named exactly as above; however, if different names are used, those names must be specified as arguments to the function. Please see below for more detail regarding how to specify those names and for more detail regarding the expectations of the function regarding the columns in the data frame.

Usage

```
oogetdoublemaxcombocutoff(
  oodataframe,
  oolistfunctionweightasafunctionofstminus = base::list(function(oodoublestminus) {
    base::return(1) }),
  oostringorsymbolid = "id",
  oostringorsymboltreated = "treated",
  oostringorsymbolAtime = "Atime",
  oostringorsymbolBtime = "Btime",
  oostringorsymbolBobserved = "Bobserved",
  oostringorsymbolCtime = "Ctime",
  oostringorsymbolCobserved = "Cobserved",
  oodoublealpha = 0.025,
  oointnmaxiter = 25L,
  oodoublelower = -10,
  oodoubleupper = 10,
  oodoublefscaletolerance = 1e-05
)
```

Arguments

- oodataframe** A data frame containing time-to-event data from two different arms.
- oolistfunctionweightasafunctionofstminus**
A list of one or more weighting functions. Each weighting function should take as input the pooled Kaplan-Meier estimate of survival across the two arms (i.e., a double between 0 and 1, inclusive) and output a single weight (i.e., a double, typically greater than or equal to zero). Defaults to the following list of a single weighting function that is identically 1, which results in the usual standardized log-rank test statistic: `base::list(function(oodoublestminus){ base::return(1) })`
- oostringorsymbolid**
The name of the column in the supplied data frame for the id variable. Defaults to "id". The id variable should take a unique value for each subject. The column with this name in the data frame can be an integer vector with a different integer for each subject; the function will most likely also work if the column is a character vector with a different value for each subject.
- oostringorsymboltreated**
The name of the column in the supplied data frame for a treatment indicator variable. Defaults to "treated". The treatment indicator variable is a variable indicating which of the two arms the subject is in. The column with this name in the data frame should be an integer vector that takes only the value 0L (e.g., for placebo) or the value 1L (e.g., for a new drug or therapy). The test statistic that is returned by this function will generally be for a test that the arm represented by subjects with treatment indicator 1L is superior to the arm represented by subjects with treatment indicator 0L (i.e., a one-sided test of superiority of arm 1 over arm 0).

- `oostringorsymbolAtime` The name of the column in the supplied data frame for the Atime variable. Defaults to "Atime". Atime is the absolute time the subject enters the study. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.
- `oostringorsymbolBtime` The name of the column in the supplied data frame for the Btime variable. Defaults to "Btime". Btime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.
- `oostringorsymbolBobserved` The name of the column in the supplied data frame for the Bobserved variable. Defaults to "Bobserved". Bobserved should be TRUE if the event was observed. The column with this name in the data frame can be a logical vector; the function will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).
- `oostringorsymbolCtime` The name of the column in the supplied data frame for the Ctime variable. Defaults to "Ctime". Ctime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.
- `oostringorsymbolCobserved` The name of the column in the supplied data frame for the Cobserved variable. Defaults to "Cobserved". Cobserved should be TRUE if censoring occurred (and so the event was not observed). The column with this name in the data frame can be a logical vector; the function will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).
- `oodoublealpha` A double that is the user-specified nominal type I error rate. Defaults to 0.025. This value should be strictly between 0 and 1.
- `oointnmaxiter` An integer that is maximum number of iterations used by the bisection method while finding the cutoff. Defaults to 25L. This value can be increased to 50L, 100L, or 200L if greater accuracy is desired while finding the cutoff; anything greater than that would typically not be needed.
- `oodoublelower` A double that is the lower bound of the interval within which the cutoff is searched for. Defaults to -10.
- `oodoubleupper` A double that is the upper bound of the interval within which the cutoff is searched for. Defaults to 10.
- `oodoublefscaletolerance` A double that is related to early stopping of the bisection method; if the bisection method finds a cutoff value for which the absolute value of the difference between the nominal type I error rate when using that cutoff value and the user-specified nominal type I error rate is less than or equal to `oodoublefscaletolerance`, the bisection method will stop and return that cutoff value. Defaults to 0.00001. This value can be decreased to a smaller strictly positive value if greater accuracy is desired while finding the cutoff.

Value

A double

Examples

```
# -----
# Example 1: Usage on a single deterministic dataset in which the drug halves
# the hazard at all times (i.e., a proportional hazards situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250
oodoublerateactive=0.125

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
    rate=oodoublerateplacebo
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateactive
  )
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
  treated=oovecinttreated,
  Atime=oovecdoubletAabsolute,
  Btime=oovecdoubletminBvsC,
  Bobserved=oovecboolobservedB,
  Ctime=oovecdoubletminBvsC,
  Cobserved=oovecboolobservedC)
```

```

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #test statistic 2.92
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #p-value 0.0017
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.92 > 1.96), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #test statistic 2.83
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #p-value 0.0023
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.83 > 1.96), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #test statistic 2.71
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10

```

```

) #p-value 0.0033
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.71 > 1.96), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #test statistic 2.92, i.e., just the maximum of 2.92 (from the log-rank test statistic) and 2.83
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #p-value 0.0028
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01,
  oodoublealpha = 0.025
) #cutoff of 2.13 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.92 > 2.13), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #test statistic 2.92, i.e., just the maximum of 2.92 (from the log-rank test statistic), 2.83
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 2.71 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #p-value 0.0032
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10,

```

```

    oodoublealpha = 0.025,
    oointnmaxiter = 100L
  ) #cutoff of 2.19 for the max-combo test statistic
  #the max-combo test statistic exceeds the cutoff (since 2.92 > 2.19), so you can declare that
  #the survival curves in the two arms are statistically significantly different at the 0.025 level

# -----
# Example 2: Usage on a single deterministic dataset in which the drug delays
# the event by exactly one month for each subject (i.e., an early treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
    rate=oodoublerateplacebo
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateplacebo
  ) + 1.0 #note the addition of 1.0 month time to event here for the active arm
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
  treated=oovecinttreated,
  Atime=oovecdoubletAabsolute,
  Btime=oovecdoubletminBvsC,
  Bobserved=oovecboolobservedB,
  Ctime=oovecdoubletminBvsC,
  Cobserved=oovecboolobservedC)

#standardized log-rank test statistic

```

```

oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #test statistic 1.66
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #p-value 0.05
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic does not exceed the cutoff (since 1.66 < 1.96), so you fail to reject
#that the survival curves in the two arms are the same at the 0.025 level

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #test statistic 0.53
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #p-value 0.30
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic does not exceed the cutoff (since 0.53 < 1.96), so you fail to reject
#that the survival curves in the two arms are the same at the 0.025 level

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #test statistic 2.07
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #p-value 0.02

```

```

maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.07 > 1.96), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #test statistic 1.66, i.e., just the maximum of 1.66 (from the log-rank test statistic) and 0.53
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #p-value 0.07
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01,
  oodoublealpha = 0.025
) #cutoff of 2.13 for the max-combo test statistic
#the max-combo test statistic does not exceed the cutoff (since 1.66 < 2.13), so you fail to reject
#that the survival curves in the two arms are the same at the 0.025 level

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #test statistic 2.07, i.e., just the maximum of 1.66 (from the log-rank test statistic), 0.53
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 2.07 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #p-value 0.03
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10,
  oodoublealpha = 0.025,

```

```

oointnmaxiter = 100L
) #cutoff of 2.20 for the max-combo test statistic
#the max-combo test statistic does not exceed the cutoff (since 2.06 < 2.20), so you fail to reject
#that the survival curves in the two arms are the same at the 0.025 level

# -----
# Example 3: Usage on a single deterministic dataset in which subjects in the placebo arm all have
# the event after being on the study for 1.2 months (i.e., a delayed treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateactive=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  base::ifelse(
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
      rate=oodoublerateactive
    ) <= 1.2,
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
      rate=oodoublerateactive
    ),
    1.2
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateactive
  )
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
  treated=oovecinttreated,
  Atime=oovecdoubletAabsolute,

```

```

Btime=oovecdoubletminBvsC,
Bobserved=oovecboolobservedB,
Ctime=oovecdoubletminBvsC,
Cobserved=oovecboolobservedC)

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #test statistic 1.55
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #p-value 0.06
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic does not exceed the cutoff (since 1.55 < 1.96), so you fail to reject
#that the survival curves in the two arms are the same at the 0.025 level

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #test statistic 2.28
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #p-value 0.01
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.28 > 1.96), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,

```

```

  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #test statistic 1.35
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #p-value 0.09
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10,
  oodoublealpha = 0.025
) #cutoff of 1.96 for the max-combo test statistic
#the max-combo test statistic does not exceed the cutoff (since 1.35 < 1.96), so you fail to reject
#that the survival curves in the two arms are the same at the 0.025 level

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #test statistic 2.28, i.e., just the maximum of 1.55 (from the log-rank test statistic) and 2.28
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #p-value 0.02
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01,
  oodoublealpha = 0.025
) #cutoff of 2.13 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.28 > 2.13), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #test statistic 2.28, i.e., just the maximum of 1.55 (from the log-rank test statistic), 2.28
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 1.35 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10

```

```

) #p-value 0.02
maxcombo::oogetdoublemaxcombocutoff(
  oodataframe = oodataframe,
  oolistfunctionweightasfunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10,
  oodoublealpha = 0.025,
  oointnmaxiter = 200L
) #cutoff of 2.15 for the max-combo test statistic
#the max-combo test statistic exceeds the cutoff (since 2.28 > 2.15), so you can declare that
#the survival curves in the two arms are statistically significantly different at the 0.025 level

```

```
oogetdoublemaxcombocutoffgroupsequential
```

Compute a cutoff value for a max-combo test statistic in a group sequential setting

Description

This function computes a cutoff value for a max-combo test statistic that is used in a group sequential setting. For example, consider the situation of a clinical trial with a planned interim analysis 2 years after the start of the clinical trial and a planned final analysis 4 years after the start of the clinical trial that plans to control the type I error rate across the interim and final analyses at the 0.025 level. Suppose the interim analysis took place 2 years after a clinical trial began, and at the time of that interim analysis, equality of the true survival curves for the two arms was tested using a max-combo test statistic based on the log-rank test statistic and the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function using a nominal type I error rate of $0.06 \times 0.025 = 0.0015$, which is what might be done in a study using the typical and conservative O'Brien-Fleming-like type I error spending, and further suppose that that max-combo test failed to reject equality of the survival curves for the two arms. Suppose that now 4 years have elapsed since the start of the study, so the final analysis is now taking place. Suppose that, as per the study protocol, the final analysis uses a max-combo test statistic based on the log-rank test statistic, the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function, *and* the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function. This is a scenario in which this function might be used, e.g., to compute a cutoff value for the max-combo test statistic for the final analysis; if the max-combo test statistic for the final analysis exceeds the cutoff returned by this function, then the study can declare that the survival curves in the two arms are statistically significantly different according to its planned analysis (note that in this case, the `oodoublealpha` increment argument should be specified to be $0.94 \times 0.025 = 0.0235$).

Please note that this function allows for the possibility that zero, one, two, or more interim analyses have occurred. Please also note that the arguments to this function expect only data and information related to the current analysis and previous analyses; this function does not expect any information in its arguments related to future planned analyses or planned future type I error spending. Also please note that, in a real clinical study, the timing of the analyses would most likely be based on the number of events that have occurred, possibly in addition to considerations related to elapsed time since the beginning of the study (e.g., "the interim analysis will take place after 50 events have occurred, and the final analysis will take place after 100 events have occurred or 4 years have elapsed

since the start of the study, whichever occurs first"); this function is compatible with such scenarios. If the true survival curves from the two arms are not at all different, then the probability, before performing the experiment, that at least one of the multiple max-combo statistics at the respective multiple analysis timepoints will be greater than its respective cutoff value for that timepoint returned by this function, is less than or approximately equal to (but typically approximately equal to) the nominal type I error rate for the group sequential experiment (e.g., 0.025 for a confirmatory clinical trial).

A max-combo test statistic is the maximum of multiple standardized weighted log-rank test statistics. The user of the function is expected to provide the function with a *list* of data frames containing the time-to-event data for the two arms; this list should contain one data frame for each previous analysis that has occurred and one data frame for the current analysis, in order, from earliest analysis timepoint up to the current analysis timepoint. Each such data frame is expected to be in a particular format (see more details below). Additionally, the user of the function is expected to provide a *list* of lists of one or more weighting functions to use for the respective one or more weighted log-rank tests for each analysis timepoint; in particular this function only allows the user to specify a weighting function that takes as input the Kaplan-Meier estimate for the survival curve obtained by pooling both of the two arms (a double that is between 0 and 1, inclusive) and provides as output a weight (a double that is typically greater than or equal to zero); note that the Fleming-Harrington class of weighting functions falls within what can be specified in this way. For each list of weighting functions for a previous analysis timepoint, the list should contain the actual weighting functions that were used at that analysis timepoint. The list of weighting functions for the current analysis timepoint should contain the weighting functions that were planned to be used at the current analysis timepoint.

Each supplied data frame is expected to have one row for each subject. Each data frame should have the following columns:

- `id` - a variable that takes a unique value for each subject
- `treated` - should be 0 for subjects in the control arm and 1 for subjects in the experimental arm.
- `Atime` - the absolute time the subject enters the study
- `Btime` - the absolute time of an event or censoring, whichever comes first
- `Bobserved` - should be TRUE if the event was observed
- `Ctime` - the absolute time of an event or censoring, whichever comes first
- `Cobserved` - should be TRUE if censoring occurred (and so the event was not observed)

The columns in the data frame do not have to be named exactly as above; however, if different names are used, those names must be specified as arguments to the function, and those names should be the same for each supplied data frame in the list of data frames supplied to the function. Please see below for more detail regarding how to specify those names and for more detail regarding the expectations of the function regarding the columns in the data frames.

Usage

```
oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint,
  oolistlistweightingfunctionsbytimepoint,
  oovecdoublecutoffsUsedInEachPreviousTimepoint,
```

```

oodoublealphaincrement,
oostingorsymbolid = "id",
oostingorsymboltreated = "treated",
oostingorsymbolAtime = "Atime",
oostingorsymbolBtime = "Btime",
oostingorsymbolBobserved = "Bobserved",
oostingorsymbolCtime = "Ctime",
oostingorsymbolCobserved = "Cobserved",
oointnmaxiter = 25L,
oodoublelower = -10,
oodoubleupper = 10,
oodoublefscaletolerance = 1e-05,
oointnrep = 100000L
)

```

Arguments

`oolistdataframesbytimepoint`

A list of one or more data frames containing time-to-event data from two different arms. This list should have one data frame for each analysis timepoint, in order from the earliest analysis timepoint up to the current analysis timepoint. Each such supplied data frame for an analysis timepoint previous to the current analysis timepoint should contain the actual data that was used at that previous analysis timepoint.

`oolistlistweightingfunctionsbytimepoint`

A list with one element for each analysis timepoint, in order from the earliest analysis timepoint up to the current analysis timepoint. Each element should be itself a list of one or more weighting functions. Each such element for an analysis timepoint previous to the current analysis timepoint should contain the actual weighting functions that were used at that previous analysis timepoint. Each weighting function should take as input the pooled Kaplan-Meier estimate of survival across the two arms (i.e., a double between 0 and 1, inclusive) and output a single weight (i.e., a double, typically greater than or equal to zero).

`oovecdoublecutoffsUsedInEachPreviousTimepoint`

A vector of doubles with one element for each *previous* analysis timepoint. Each such element should contain the actual cutoff value (e.g., as was returned by a previous invocation of this function) that was used at that previous analysis timepoint. Please note that the length of this argument should be exactly 1 less than the length of `oolistdataframesbytimepoint` and `oolistlistweightingfunctionsbytimepoint`, which should have exactly the same length. If there were no previous analysis timepoints (e.g., the current analysis timepoint is the first planned interim analysis timepoint), then this argument should be specified to be `base::vector(mode="double", length=0L)`.

`oodoublealphaincrement`

A double that is the user-specified nominal type I error rate for the *current* analysis timepoint. In a group sequential experiment, the nominal type I error of the experiment is the *sum* of the nominal type I error rates for each respective analysis timepoint. This value should be strictly greater than 0, and the sum of this value for the current analysis timepoint and any previous analysis timepoints should be strictly less than 1.

oostringorsymbolid

The name of the column in the supplied data frames for the id variable. Defaults to "id". The id variable should take a unique value for each subject. The column with this name in the data frames can be an integer vector with a different integer for each subject; the function will most likely also work if the column is a character vector with a different value for each subject.

oostringorsymboltreated

The name of the column in the supplied data frames for a treatment indicator variable. Defaults to "treated". The treatment indicator variable is a variable indicating which of the two arms the subject is in. The column with this name in the data frames should be an integer vector that takes only the value 0L (e.g., for placebo) or the value 1L (e.g., for a new drug or therapy). The test statistic that is returned by this function will generally be for a test that the arm represented by subjects with treatment indicator 1L is superior to the arm represented by subjects with treatment indicator 0L (i.e., a one-sided test of superiority of arm 1 over arm 0).

oostringorsymbolAtime

The name of the column in the supplied data frames for the Atime variable. Defaults to "Atime". Atime is the absolute time the subject enters the study. The column with this name in the data frames can be vector of doubles; the function will most likely also work if the column is a vector of integers.

oostringorsymbolBtime

The name of the column in the supplied data frames for the Btime variable. Defaults to "Btime". Btime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frames can be vector of doubles; the function will most likely also work if the column is a vector of integers.

oostringorsymbolBobserved

The name of the column in the supplied data frames for the Bobserved variable. Defaults to "Bobserved". Bobserved should be TRUE if the event was observed. The column with this name in the data frames can be a logical vector; the function will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).

oostringorsymbolCtime

The name of the column in the supplied data frames for the Ctime variable. Defaults to "Ctime". Ctime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frames can be vector of doubles; the function will most likely also work if the column is a vector of integers.

oostringorsymbolCobserved

The name of the column in the supplied data frames for the Cobserved variable. Defaults to "Cobserved". Cobserved should be TRUE if censoring occurred (and so the event was not observed). The column with this name in the data frame can be a logical vector; the function will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).

oointnmaxiter

An integer that is maximum number of iterations used by the bisection method while finding the cutoff. Defaults to 25L. This value can be increased to 50L, 100L, or 200L if greater accuracy is desired while finding the cutoff; anything greater than that would typically not be needed.

- `oodoublelower` A double that is the lower bound of the interval within which the cutoff is searched for. Defaults to -10.
- `oodoubleupper` A double that is the upper bound of the interval within which the cutoff is searched for. Defaults to 10.
- `oodoublefscaletolerance`
A double that is related to early stopping of the bisection method; if the bisection method finds a cutoff value for which the absolute value of the difference between the nominal type I error rate when using that cutoff value and the user-specified nominal type I error rate is less than or equal to `oodoublefscaletolerance`, the bisection method will stop and return that cutoff value. Defaults to 0.00001. This value can be decreased to a smaller strictly positive value if greater accuracy is desired while finding the cutoff.
- `oointnrep` This function may use repeated simulations to estimate the type I error rate; this argument controls how many simulations are used to do this. Defaults to 100000L. There is typically not much need to change this argument from its default value.

Value

A double

Examples

```
#this function is used in the examples below to create synthetic earlier looks
#at the simulated datasets.
oogetdataframeearlierlookattime=function(oodataframe,oodoubletime)
{
  dplyr::mutate(
    dplyr::filter(oodataframe,Atime <= oodoubletime),
    Bobserved = base::ifelse(Btime <= oodoubletime,Bobserved,FALSE),
    Cobserved = base::ifelse(Ctime <= oodoubletime,Cobserved,FALSE),
    Btime = base::pmin(Btime,oodoubletime),
    Ctime = base::pmin(Ctime,oodoubletime)
  )
}

# -----
# Example 1: Usage on a single deterministic dataset in which the drug halves
# the hazard at all times (i.e., a proportional hazards situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250
oodoublerateactive=0.125

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
```

```

    base::rep(1L,length.out=ooointnparticipantsactive)
  )
  oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
    base::seq(from=0.0,to=2.0,length.out=ooointnparticipantsplacebo),
    base::seq(from=0.0,to=2.0,length.out=ooointnparticipantsactive)
  )
  #the duration of time from when the subject enters the study until the subject experiences the event
  oovecdoubletAtoB=c(
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=ooointnparticipantsplacebo),
      rate=oodoublerateplacebo
    ),
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=ooointnparticipantsactive),
      rate=oodoublerateactive
    )
  )
  oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
  #the final analysis takes place at absolute time 6.0 months, and no other
  #censoring (e.g., dropout) occurs
  oovecdoubletCabsolute=6.0
  oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
  oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
  oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

  oodataframe=dplyr::tibble(id=1L:ooointnparticipants,
    treated=oovecinttreated,
    Atime=oovecdoubletAabsolute,
    Btime=oovecdoubletminBvsC,
    Bobserved=oovecboolobservedB,
    Ctime=oovecdoubletminBvsC,
    Cobserved=oovecboolobservedC)

  oodataframeinterimanalysis=oogetdataframeearlierlookattime(
    oodataframe = oodataframe,
    oodoubletime = 3.0 #the interim analysis takes place at absolute time 3.0 months
  )
  oodataframefinalanalysis=oodataframe #the final analysis takes place at absolute time 6.0 months

  #plan to use the standardized log-rank test statistic at both the interim and the final analysis
  oolistlistweightingfunctionsbytimepoint=base::list(
    interim=base::list(
      logrank=function(stminus){ base::return(1.0) }
    ),
    final=base::list(
      logrank=function(stminus){ base::return(1.0) }
    )
  )
  maxcombo::oogetdoublemaxcombostatistic(
    oodataframe = oodataframeinterimanalysis,
    oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
  )

```

```

) #max-combo test statistic 2.36 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.36 < 2.97)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.92 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 1.96 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis exceeds the cutoff (since 2.92 > 1.96),
#so you can declare that the survival curves in the two arms are statistically significantly
#different at the 0.025 level

#plan to use the standardized log-rank test statistic at the interim analysis and to use the
#max-combo test statistic based on the standardized log-rank test statistic and the weighted
#log-rank test statistic with Fleming-Harrington 0-1 weighting function, which places greater
#weight on later times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.36 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)

```

```

oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.36 < 2.97)
#so the experiment continues until the final analysis.

```

```

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.92 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.12 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis exceeds the cutoff (since 2.92 > 2.12),
#so you can declare that the survival curves in the two arms are statistically significantly
#different at the 0.025 level

```

```

#plan to use the standardized log-rank test statistic at the interim analysis and to use the
#max-combo test statistic based on the standardized log-rank test statistic, the weighted
#log-rank test statistic with Fleming-Harrington 0-1 weighting function, which places greater
#weight on later times, and the weighted log-rank test statistic with Fleming-Harrington 1-0
#weighting function, which places greater weight on earlier times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(

```

```

  interim=base::list(
    logrank=function(stminus){ base::return(1.0) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.36 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.36 < 2.97)
#so the experiment continues until the final analysis.

```

```

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.92 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis, oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.19 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis exceeds the cutoff (since 2.92 > 2.19),
#so you can declare that the survival curves in the two arms are statistically significantly
#different at the 0.025 level

#plan to use the standardized log-rank test statistic and the weighted log-rank test
#statistic with Fleming-Harrington 0-1 weighting function at the interim analysis and to use
#the max-combo test statistic based on the standardized log-rank test statistic, the weighted
#log-rank test statistic with Fleming-Harrington 0-1 weighting function, which places greater weight
#on later times, and the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting
#function, which places greater weight on earlier times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.56 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 3.13 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.56 < 3.13)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
)

```

```

) #max-combo test statistic 2.92 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.20 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis exceeds the cutoff (since 2.92 > 2.20),
#so you can declare that the survival curves in the two arms are statistically significantly
#different at the 0.025 level

# -----
# Example 2: Usage on a single deterministic dataset in which the drug delays
# the event by exactly one month for each subject (i.e., an early treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
    rate=oodoublerateplacebo
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateplacebo
  ) + 1.0 #note the addition of 1.0 month time to event here for the active arm
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
  treated=oovecinttreated,

```

```

        Atime=oovecdoubletAabsolute,
        Btime=oovecdoubletminBvsC,
        Bobserved=oovecboolobservedB,
        Ctime=oovecdoubletminBvsC,
        Cobserved=oovecboolobservedC)

oodataframeinterimanalysis=oogetdataframeearlierlookattime(
  oodataframe = oodataframe,
  oodoubletime = 3.0 #the interim analysis takes place at absolute time 3.0 months
)
oodataframefinalanalysis=oodataframe #the final analysis takes place at absolute time 6.0 months

#plan to use the standardized log-rank test statistic at both the interim and the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.51 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  ooecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.51 < 2.97)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 1.66 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  ooecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 1.96 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis does not exceed the cutoff
#(since 1.66 < 1.96), so you fail to reject that the survival curves in the two arms are
#the same at the 0.025 level

```

```

#plan to use the standardized log-rank test statistic at the interim analysis and to use the
#max-combo test statistic based on the standardized log-rank test statistic and the weighted
#log-rank test statistic with Fleming-Harrington 0-1 weighting function, which places greater
#weight on later times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.51 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.51 < 2.97)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 1.66 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.12 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis does not exceed the cutoff
#(since 1.66 < 2.12), so you fail to reject that the survival curves in the two arms are
#the same at the 0.025 level

#plan to use the standardized log-rank test statistic at the interim analysis and to use the
#max-combo test statistic based on the standardized log-rank test statistic, the weighted
#log-rank test statistic with Fleming-Harrington 0-1 weighting function, which places greater
#weight on later times, and the weighted log-rank test statistic with Fleming-Harrington 1-0

```

```

#weighting function, which places greater weight on earlier times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.51 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  ooecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.51 < 2.97)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.07 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  ooecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.19 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis does not exceed the cutoff
#(since 2.07 < 2.19), so you fail to reject that the survival curves in the two arms are
#the same at the 0.025 level

#plan to use the standardized log-rank test statistic and the weighted log-rank test
#statistic with Fleming-Harrington 0-1 weighting function, which places greater weight on
#later times, at the interim analysis and to use the max-combo test statistic based on
#the standardized log-rank test statistic, the weighted log-rank test statistic with
#Fleming-Harrington 0-1 weighting function, which places greater weight on later times,
#and the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting
#function, which places greater weight on earlier times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(

```

```

    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  )
)
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafuncofstinus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.51 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
)
oodoublecutoffforinterimanalysis #cutoff of 3.13 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.51 < 3.13)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafuncofstinus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.07 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
)
oodoublecutoffforfinalanalysis #cutoff of 2.20 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis does not exceed the cutoff
#(since 2.07 < 2.20), so you fail to reject that the survival curves in the two arms are
#the same at the 0.025 level

#plan to use the standardized log-rank test statistic and the weighted log-rank test
#statistic with Fleming-Harrington 1-0 weighting function, which places greater weight on
#earlier times, at the interim analysis and to use the max-combo test statistic based on
#the standardized log-rank test statistic, the weighted log-rank test statistic with
#Fleming-Harrington 1-0 weighting function, which places greater weight on earlier times,
#at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  ),
  final=base::list(

```

```

    logrank=function(stminus){ base::return(1.0) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 2.71 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 3.02 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 2.71 < 3.02)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.07 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.06 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis exceeds the cutoff (since 2.07 > 2.06),
#so you can declare that the survival curves in the two arms are statistically significantly
#different at the 0.025 level

# -----
# Example 3: Usage on a single deterministic dataset in which subjects in the placebo arm all have
# the event after being on the study for 1.2 months (i.e., a delayed treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateactive=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),

```

```

base::seq(from=0.0,to=2.0,length.out=ooointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  base::ifelse(
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=ooointnparticipantsplacebo),
      rate=oodoublerateactive
    ) <= 1.2,
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=ooointnparticipantsplacebo),
      rate=oodoublerateactive
    ),
    1.2
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=ooointnparticipantsactive),
    rate=oodoublerateactive
  )
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:ooointnparticipants,
                          treated=oovecinttreated,
                          Atime=oovecdoubletAabsolute,
                          Btime=oovecdoubletminBvsC,
                          Bobserved=oovecboolobservedB,
                          Ctime=oovecdoubletminBvsC,
                          Cobserved=oovecboolobservedC)

oodataframeinterimanalysis=oogetdataframeearlierlookattime(
  oodataframe = oodataframe,
  oodoubletime = 3.0 #the interim analysis takes place at absolute time 3.0 months
)
oodataframefinalanalysis=oodataframe #the final analysis takes place at absolute time 6.0 months

#plan to use the standardized log-rank test statistic at both the interim and the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) }
  )
)
)
maxcombo::oogetdoublemaxcombostatistic(

```

```

    oodataframe = oodataframeinterimanalysis,
    oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 1.24 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 1.24 < 2.97)
#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 1.55 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 1.96 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis does not exceed the cutoff
#(since 1.55 < 1.96), so you fail to reject that the survival curves in the two arms are
#the same at the 0.025 level

#plan to use the standardized log-rank test statistic at the interim analysis and to use the
#max-combo test statistic based on the standardized log-rank test statistic and the weighted
#log-rank test statistic with Fleming-Harrington 0-1 weighting function, which places greater
#weight on later times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 1.24 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),

```

```

    oodoublealphaincrement = 0.06*0.025
  )
  oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
  #the interim analysis
  #the max-combo test statistic at the interim analysis does not exceed the cutoff (since 1.24 < 2.97)
  #so the experiment continues until the final analysis.

  maxcombo::oogetdoublemaxcombostatistic(
    oodataframe = oodataframefinalanalysis,
    oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
  ) #max-combo test statistic 2.28 at the final analysis
  oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
    oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
    oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
    oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
    oodoublealphaincrement = 0.025 - 0.06*0.025
  )
  oodoublecutoffforfinalanalysis #cutoff of 2.12 for the max-combo test statistic at
  #the final analysis
  #the max-combo test statistic at the final analysis exceeds the cutoff (since 2.28 > 2.12),
  #so you can declare that the survival curves in the two arms are statistically significantly
  #different at the 0.025 level

  #plan to use the standardized log-rank test statistic at the interim analysis and to use the
  #max-combo test statistic based on the standardized log-rank test statistic, the weighted
  #log-rank test statistic with Fleming-Harrington 0-1 weighting function, which places greater
  #weight on later times, and the weighted log-rank test statistic with Fleming-Harrington 1-0
  #weighting function, which places greater weight on earlier times, at the final analysis
  oolistlistweightingfunctionsbytimepoint=base::list(
    interim=base::list(
      logrank=function(stminus){ base::return(1.0) }
    ),
    final=base::list(
      logrank=function(stminus){ base::return(1.0) },
      flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
      flemingharrington10=function(stminus){ base::return(stminus) }
    )
  )
  maxcombo::oogetdoublemaxcombostatistic(
    oodataframe = oodataframeinterimanalysis,
    oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
  ) #max-combo test statistic 1.24 at the interim analysis
  oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
    oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
    oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
    oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
    oodoublealphaincrement = 0.06*0.025
  )
  oodoublecutoffforinterimanalysis #cutoff of 2.97 for the max-combo test statistic at
  #the interim analysis
  #the max-combo test statistic at the interim analysis does not exceed the cutoff (since 1.24 < 2.97)

```

```

#so the experiment continues until the final analysis.

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.28 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.16 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis exceeds the cutoff (since 2.28 > 2.16),
#so you can declare that the survival curves in the two arms are statistically significantly
#different at the 0.025 level

#plan to use the standardized log-rank test statistic and the weighted log-rank test
#statistic with Fleming-Harrington 0-1 weighting function, which places greater weight on
#later times, at the interim analysis and to use the max-combo test statistic based on
#the standardized log-rank test statistic, the weighted log-rank test statistic with
#Fleming-Harrington 0-1 weighting function, which places greater weight on later times,
#and the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting
#function, which places greater weight on earlier times, at the final analysis
oolistlistweightingfunctionsbytimepoint=base::list(
  interim=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
  ),
  final=base::list(
    logrank=function(stminus){ base::return(1.0) },
    flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  )
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframeinterimanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[1L]]
) #max-combo test statistic 1.88 at the interim analysis
oodoublecutoffforinterimanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:1L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = base::vector(mode="double",length=0L),
  oodoublealphaincrement = 0.06*0.025
)
oodoublecutoffforinterimanalysis #cutoff of 3.12 for the max-combo test statistic at
#the interim analysis
#the max-combo test statistic at the interim analysis does not exceed the cutoff (since 1.88 < 3.12)
#so the experiment continues until the final analysis.

```

```

maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframefinalanalysis,
  oolistfunctionweightasafunctionofstminus = oolistlistweightingfunctionsbytimepoint[[2L]]
) #max-combo test statistic 2.28 at the final analysis
oodoublecutoffforfinalanalysis=maxcombo::oogetdoublemaxcombocutoffgroupsequential(
  oolistdataframesbytimepoint = base::list(oodataframeinterimanalysis,oodataframefinalanalysis),
  oolistlistweightingfunctionsbytimepoint = oolistlistweightingfunctionsbytimepoint[1L:2L],
  oovecdoublecutoffsUsedInEachPreviousTimepoint = c(oodoublecutoffforinterimanalysis),
  oodoublealphaincrement = 0.025 - 0.06*0.025
)
oodoublecutoffforfinalanalysis #cutoff of 2.16 for the max-combo test statistic at
#the final analysis
#the max-combo test statistic at the final analysis exceeds the cutoff (since 2.28 > 2.16),
#so you can declare that the survival curves in the two arms are statistically significantly
#different at the 0.025 level

```

```
oogetdoublemaxcombostatpvalue
```

Compute the p-value for the max-combo test statistic at a single time-point

Description

This function computes the p-value for the max-combo test statistic for comparing times to an event from two different arms at a single timepoint; please note that this is *not* a p-value for a group sequential test procedure involving two or more timepoints, but rather a p-value for a scenario in which there is only one planned analysis timepoint. The max-combo test statistic is the maximum of multiple standardized weighted log-rank test statistics. The user of the function is expected to provide the function with a data frame containing the time-to-event data for the two arms; additionally, this data frame is expected to be in a particular format (see more details below). Additionally, the user of the function is expected to provide a list of one or more weighting functions to use for the respective one or more weighted log-rank tests; in particular this function only allows the user to specify a weighting function that takes as input the Kaplan-Meier estimate for the survival curve obtained by pooling both of the two arms (a double that is between 0 and 1, inclusive) and provides as output a weight (a double that is typically greater than or equal to zero); note that the Fleming-Harrington class of weighting functions falls within what can be specified in this way. If no list of weighting functions is provided, a single weighting function that is identically 1 is used, which means that the test statistic that is used is exactly equal to the usual standardized log-rank test statistic.

The supplied data frame is expected to have one row for each subject. The data frame should have the following columns:

- `id` - a variable that takes a unique value for each subject
- `treated` - should be 0 for subjects in the control arm and 1 for subjects in the experimental arm.
- `Atime` - the absolute time the subject enters the study

- Btime - the absolute time of an event or censoring, whichever comes first
- Boberved - should be TRUE if the event was observed
- Ctime - the absolute time of an event or censoring, whichever comes first
- Coberved - should be TRUE if censoring occurred (and so the event was not observed)

The columns in the data frame do not have to be named exactly as above; however, if different names are used, those names must be specified as arguments to the function. Please see below for more detail regarding how to specify those names and for more detail regarding the expectations of the function regarding the columns in the data frame.

Usage

```
oogetdoublemaxcombotestpvalue(
  oodataframe,
  oolistfunctionweightasafunctionofstminus = base::list(function(oodoublestminus) {
    base::return(1) }),
  oostingorsymbolid = "id",
  oostingorsymboltreated = "treated",
  oostingorsymbolAtime = "Atime",
  oostingorsymbolBtime = "Btime",
  oostingorsymbolBoberved = "Boberved",
  oostingorsymbolCtime = "Ctime",
  oostingorsymbolCoberved = "Coberved",
  oodoublemaxcomboteststatistic = NULL
)
```

Arguments

oodataframe A data frame containing time-to-event data from two different arms.

oolistfunctionweightasafunctionofstminus

A list of one or more weighting functions. Each weighting function should take as input the pooled Kaplan-Meier estimate of survival across the two arms (i.e., a double between 0 and 1, inclusive) and output a single weight (i.e., a double, typically greater than or equal to zero). Defaults to the following list of a single weighting function that is identically 1, which results in the usual standardized log-rank test statistic: `base::list(function(oodoublestminus){ base::return(1) })`

oostingorsymbolid

The name of the column in the supplied data frame for the id variable. Defaults to "id". The id variable should take a unique value for each subject. The column with this name in the data frame can be an integer vector with a different integer for each subject; the function will most likely also work if the column is a character vector with a different value for each subject.

oostingorsymboltreated

The name of the column in the supplied data frame for a treatment indicator variable. Defaults to "treated". The treatment indicator variable is a variable indicating which of the two arms the subject is in. The column with this name in the data frame should be an integer vector that takes only the value 0L (e.g.,

for placebo) or the value 1L (e.g., for a new drug or therapy). The test statistic that is used by this function will generally be for a test that the arm represented by subjects with treatment indicator 1L is superior to the arm represented by subjects with treatment indicator 0L (i.e., a one-sided test of superiority of arm 1 over arm 0).

`oostringorsymbolAtime`

The name of the column in the supplied data frame for the Atime variable. Defaults to "Atime". Atime is the absolute time the subject enters the study. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.

`oostringorsymbolBtime`

The name of the column in the supplied data frame for the Btime variable. Defaults to "Btime". Btime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.

`oostringorsymbolBobserved`

The name of the column in the supplied data frame for the Bobserved variable. Defaults to "Bobserved". Bobserved should be TRUE if the event was observed. The column with this name in the data frame can be a logical vector; the function will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).

`oostringorsymbolCtime`

The name of the column in the supplied data frame for the Ctime variable. Defaults to "Ctime". Ctime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.

`oostringorsymbolCobserved`

The name of the column in the supplied data frame for the Cobserved variable. Defaults to "Cobserved". Cobserved should be TRUE if censoring occurred (and so the event was not observed). The column with this name in the data frame can be a logical vector; the function will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).

`oodoublemaxcombobetststatistic`

A value for a hypothetical max-combo test statistic that could have been observed (but was not) can be provided to this argument, in which case the p-value for that hypothetical statistic that could have been observed (but was not) is what is provided. If NULL is provided, a p-value for the max-combo test statistic that was actually observed is provided. Defaults to NULL.

Value

A double

Examples

```
# -----
# Example 1: Usage on a single deterministic dataset in which the drug halves
```

```

# the hazard at all times (i.e., a proportional hazards situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250
oodoublerateactive=0.125

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
    rate=oodoublerateplacebo
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateactive
  )
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
                          treated=oovecinttreated,
                          Atime=oovecdoubletAabsolute,
                          Btime=oovecdoubletminBvsC,
                          Bobserved=oovecboolobservedB,
                          Ctime=oovecdoubletminBvsC,
                          Cobserved=oovecboolobservedC)

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafuncofstminus = oolistweightingfunctionsJustLogrank
) #test statistic 2.92
maxcombo::oogetdoublemaxcombotestpvalue(

```

```

    oodataframe = oodataframe,
    oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #p-value 0.0017

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #test statistic 2.83
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #p-value 0.0023

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #test statistic 2.71
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #p-value 0.0033

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #test statistic 2.92, i.e., just the maximum of 2.92 (from the log-rank test statistic) and 2.83
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #p-value 0.0028

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
  flemingharrington10=function(stminus){ base::return(stminus) }
)

```

```

maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #test statistic 2.92, i.e., just the maximum of 2.92 (from the log-rank test statistic), 2.83
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 2.71 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #p-value 0.0032

# -----
# Example 2: Usage on a single deterministic dataset in which the drug delays
# the event by exactly one month for each subject (i.e., an early treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
    rate=oodoublerateplacebo
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateplacebo
  ) + 1.0 #note the addition of 1.0 month time to event here for the active arm
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
  treated=oovecinttreated,

```

```

        Atime=oovecdoubletAabsolute,
        Btime=oovecdoubletminBvsC,
        Bobserved=oovecboolobservedB,
        Ctime=oovecdoubletminBvsC,
        Cobserved=oovecboolobservedC)

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #test statistic 1.66
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #p-value 0.05

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #test statistic 0.53
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #p-value 0.30

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #test statistic 2.07
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #p-value 0.02

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatpvalue(

```

```

    oodataframe = oodataframe,
    oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #test statistic 1.66, i.e., just the maximum of 1.66 (from the log-rank test statistic) and 0.53
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #p-value 0.07

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #test statistic 2.07, i.e., just the maximum of 1.66 (from the log-rank test statistic), 0.53
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 2.07 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #p-value 0.03

# -----
# Example 3: Usage on a single deterministic dataset in which subjects in the placebo arm all have
# the event after being on the study for 1.2 months (i.e., a delayed treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateactive=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  base::ifelse(
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),

```

```

    rate=oodoublerateactive
  ) <= 1.2,
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=ooointnparticipantsplacebo),
    rate=oodoublerateactive
  ),
  1.2
),
stats::qexp(
  base::seq(from=0.0,to=0.98,length.out=ooointnparticipantsactive),
  rate=oodoublerateactive
)
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:ooointnparticipants,
                          treated=oovecinttreated,
                          Atime=oovecdoubletAabsolute,
                          Btime=oovecdoubletminBvsC,
                          Bobserved=oovecboolobservedB,
                          Ctime=oovecdoubletminBvsC,
                          Cobserved=oovecboolobservedC)

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #test statistic 1.55
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #p-value 0.06

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #test statistic 2.28
maxcombo::oogetdoublemaxcombostatpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01

```

```

) #p-value 0.01

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #test statistic 1.35
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #p-value 0.09

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #test statistic 2.28, i.e., just the maximum of 1.55 (from the log-rank test statistic) and 2.28
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #p-value 0.02

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #test statistic 2.28, i.e., just the maximum of 1.55 (from the log-rank test statistic), 2.28
# (from the weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 1.35 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)
maxcombo::oogetdoublemaxcombotestpvalue(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #p-value 0.02

```

 oogetdoublemaxcombostatistic

Compute the max-combo test statistic

Description

This function computes the max-combo test statistic for comparing times to an event from two different arms. The max-combo test statistic is the maximum of multiple standardized weighted log-rank test statistics. The user of the function is expected to provide the function with a data frame containing the time-to-event data for the two arms; additionally, this data frame is expected to be in a particular format (see more details below). Additionally, the user of the function is expected to provide a list of one or more weighting functions to use for the respective one or more weighted log-rank tests; in particular this function only allows the user to specify a weighting function that takes as input the Kaplan-Meier estimate for the survival curve obtained by pooling both of the two arms (a double that is between 0 and 1, inclusive) and provides as output a weight (a double that is typically greater than or equal to zero); note that the Fleming-Harrington class of weighting functions falls within what can be specified in this way. If no list of weighting functions is provided, a single weighting function that is identically 1 is used, which means that the test statistic that is returned is exactly equal to the usual standardized log-rank test statistic.

The supplied data frame is expected to have one row for each subject. The data frame should have the following columns:

- `id` - a variable that takes a unique value for each subject
- `treated` - should be 0 for subjects in the control arm and 1 for subjects in the experimental arm.
- `Atime` - the absolute time the subject enters the study
- `Btime` - the absolute time of an event or censoring, whichever comes first
- `Bobserved` - should be TRUE if the event was observed
- `Ctime` - the absolute time of an event or censoring, whichever comes first
- `Cobserved` - should be TRUE if censoring occurred (and so the event was not observed)

The columns in the data frame do not have to be named exactly as above; however, if different names are used, those names must be specified as arguments to the function. Please see below for more detail regarding how to specify those names and for more detail regarding the expectations of the function regarding the columns in the data frame.

Usage

```
oogetdoublemaxcombostatistic(
  oodataframe,
  oolistfunctionweightasafunctionofstminus = base::list(function(oodoublestminus) {
    base::return(1) }),
  oostringorsymbolid = "id",
  oostringorsymboltreated = "treated",
  oostringorsymbolAtime = "Atime",
```

```

oostringorsymbolBtime = "Btime",
oostringorsymbolBobserved = "Bobserved",
oostringorsymbolCtime = "Ctime",
oostringorsymbolCobserved = "Cobserved"
)

```

Arguments

oodataframe A data frame containing time-to-event data from two different arms.

oolistfunctionweightasafunctionofstminus

A list of one or more weighting functions. Each weighting function should take as input the pooled Kaplan-Meier estimate of survival across the two arms (i.e., a double between 0 and 1, inclusive) and output a single weight (i.e., a double, typically greater than or equal to zero). Defaults to the following list of a single weighting function that is identically 1, which results in the usual standardized log-rank test statistic: `base::list(function(oodoublestminus){ base::return(1) })`

oostringorsymbolid

The name of the column in the supplied data frame for the id variable. Defaults to "id". The id variable should take a unique value for each subject. The column with this name in the data frame can be an integer vector with a different integer for each subject; the function will most likely also work if the column is a character vector with a different value for each subject.

oostringorsymboltreated

The name of the column in the supplied data frame for a treatment indicator variable. Defaults to "treated". The treatment indicator variable is a variable indicating which of the two arms the subject is in. The column with this name in the data frame should be an integer vector that takes only the value 0L (e.g., for placebo) or the value 1L (e.g., for a new drug or therapy). The test statistic that is returned by this function will generally be for a test that the arm represented by subjects with treatment indicator 1L is superior to the arm represented by subjects with treatment indicator 0L (i.e., a one-sided test of superiority of arm 1 over arm 0).

oostringorsymbolAtime

The name of the column in the supplied data frame for the Atime variable. Defaults to "Atime". Atime is the absolute time the subject enters the study. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.

oostringorsymbolBtime

The name of the column in the supplied data frame for the Btime variable. Defaults to "Btime". Btime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.

oostringorsymbolBobserved

The name of the column in the supplied data frame for the Bobserved variable. Defaults to "Bobserved". Bobserved should be TRUE if the event was observed. The column with this name in the data frame can be a logical vector; the function

will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).

oostringorsymbolCtime

The name of the column in the supplied data frame for the Ctime variable. Defaults to "Ctime". Ctime is the absolute time of an event or censoring, whichever comes first. The column with this name in the data frame can be vector of doubles; the function will most likely also work if the column is a vector of integers.

oostringorsymbolCobserved

The name of the column in the supplied data frame for the Cobserved variable. Defaults to "Cobserved". Cobserved should be TRUE if censoring occurred (and so the event was not observed). The column with this name in the data frame can be a logical vector; the function will most likely also work if the column is a vector of integers (i.e., with 1L in place of TRUE and 0L in place of FALSE).

Value

A double

Examples

```
# -----
# Example 1: Usage on a single deterministic dataset in which the drug halves
# the hazard at all times (i.e., a proportional hazards situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250
oodoublerateactive=0.125

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
    rate=oodoublerateplacebo
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateactive
  )
)
```

```

)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
                          treated=oovecinttreated,
                          Atime=oovecdoubletAabsolute,
                          Btime=oovecdoubletminBvsC,
                          Bobserved=oovecboolobservedB,
                          Ctime=oovecdoubletminBvsC,
                          Cobserved=oovecboolobservedC)

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #2.92

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #2.83

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #2.71

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
)

```

```

) #2.92, i.e., just the maximum of 2.92 (from the log-rank test statistic) and 2.83 (from the
# weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasfunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #2.92, i.e., just the maximum of 2.92 (from the log-rank test statistic), 2.83 (from the
# weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 2.71 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)

# -----
# Example 2: Usage on a single deterministic dataset in which the drug delays
# the event by exactly one month for each subject (i.e., an early treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateplacebo=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
    rate=oodoublerateplacebo
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateplacebo
  ) + 1.0 #note the addition of 1.0 month time to event here for the active arm
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)
oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)

```

```

oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
                          treated=oovecinttreated,
                          Atime=oovecdoubletAabsolute,
                          Btime=oovecdoubletminBvsC,
                          Bobserved=oovecboolobservedB,
                          Ctime=oovecdoubletminBvsC,
                          Cobserved=oovecboolobservedC)

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #1.66

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #0.53

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #2.07

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #1.66, i.e., just the maximum of 1.66 (from the log-rank test statistic) and 0.53 (from the
# weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(
  logrank=function(stminus){ base::return(1.0) },

```

```

    flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
    flemingharrington10=function(stminus){ base::return(stminus) }
  )
maxcombo::oogetdoublemaxcombostatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #2.07, i.e., just the maximum of 1.66 (from the log-rank test statistic), 0.53 (from the
# weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 2.07 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)

# -----
# Example 3: Usage on a single deterministic dataset in which subjects in the placebo arm all have
# the event after being on the study for 1.2 months (i.e., a delayed treatment effect situation)
# -----

oointnparticipants=100L
oointnparticipantsplacebo=oointnparticipants/2L
oointnparticipantsactive=oointnparticipants/2L

oodoublerateactive=0.250

oovecinttreated=c(
  base::rep(0L,length.out=oointnparticipantsplacebo),
  base::rep(1L,length.out=oointnparticipantsactive)
)
oovecdoubletAabsolute=c( #the start time, i.e., when the subject enters the study.
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsplacebo),
  base::seq(from=0.0,to=2.0,length.out=oointnparticipantsactive)
)
#the duration of time from when the subject enters the study until the subject experiences the event
oovecdoubletAtoB=c(
  base::ifelse(
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
      rate=oodoublerateactive
    ) <= 1.2,
    stats::qexp(
      base::seq(from=0.0,to=0.98,length.out=oointnparticipantsplacebo),
      rate=oodoublerateactive
    ),
    1.2
  ),
  stats::qexp(
    base::seq(from=0.0,to=0.98,length.out=oointnparticipantsactive),
    rate=oodoublerateactive
  )
)
oovecdoubletBabsolute=oovecdoubletAabsolute + oovecdoubletAtoB
#the analysis takes place at absolute time 6.0 months, and no other censoring (e.g., dropout) occurs
oovecdoubletCabsolute=6.0
oovecdoubletminBvsC=base::pmin(oovecdoubletBabsolute,oovecdoubletCabsolute)

```

```

oovecboolobservedB=(oovecdoubletBabsolute < oovecdoubletCabsolute)
oovecboolobservedC=(oovecdoubletCabsolute <= oovecdoubletBabsolute)

oodataframe=dplyr::tibble(id=1L:oointnparticipants,
                          treated=oovecinttreated,
                          Atime=oovecdoubletAabsolute,
                          Btime=oovecdoubletminBvsC,
                          Bobserved=oovecboolobservedB,
                          Ctime=oovecdoubletminBvsC,
                          Coberved=oovecboolobservedC)

#standardized log-rank test statistic
oolistweightingfunctionsJustLogrank=base::list(
  logrank=function(stminus){ base::return(1.0) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustLogrank
) #1.55

#standardized weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function,
#which places greater weight on later times
oolistweightingfunctionsJustFlemingHarrington01=base::list(
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington01
) #2.28

#standardized weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function,
#which places greater weight on earlier times
oolistweightingfunctionsJustFlemingHarrington10=base::list(
  flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsJustFlemingHarrington10
) #1.35

#the max-combo test statistic based on the first two of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01=base::list(
  logrank=function(stminus){ base::return(1.0) },
  flemingharrington01=function(stminus){ base::return(1.0 - stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus = oolistweightingfunctionsLogrankAndFlemingHarrington01
) #2.28, i.e., just the maximum of 1.55 (from the log-rank test statistic) and 2.28 (from the
# weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function)

#the max-combo test statistic based on the first three of the above
oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10=base::list(

```

```
logrank=function(stminus){ base::return(1.0) },
flemingharrington01=function(stminus){ base::return(1.0 - stminus) },
flemingharrington10=function(stminus){ base::return(stminus) }
)
maxcombo::oogetdoublemaxcomboteststatistic(
  oodataframe = oodataframe,
  oolistfunctionweightasafunctionofstminus =
  oolistweightingfunctionsLogrankAndFlemingHarrington01AndFlemingHarrington10
) #2.28, i.e., just the maximum of 1.55 (from the log-rank test statistic), 2.28 (from the
# weighted log-rank test statistic with Fleming-Harrington 0-1 weighting function), and
# 1.35 (from the weighted log-rank test statistic with Fleming-Harrington 1-0 weighting function)
```

Index

* **package**

maxcombo-package, [2](#)

maxcombo-package, [2](#)

oogetdoublemaxcombocutoff, [2](#), [3](#)

oogetdoublemaxcombocutoffgroupsequential,
[2](#), [15](#)

oogetdoublemaxcombotestpvalue, [2](#), [34](#)

oogetdoublemaxcomboteststatistic, [2](#), [44](#)