Package 'npsurvSS'

May 9, 2026

Type Package

Title Sample Size and Power Calculation for Common Non-Parametric Tests in Survival Analysis

Version 1.1.0

Author Godwin Yung [aut, cre], Yi Liu [aut]

Maintainer Godwin Yung <godwin.y.yung@gmail.com>

Description A number of statistical tests have been proposed to compare two survival curves, including the difference in (or ratio of) t-year survival, difference in (or ratio of) p-th percentile survival, difference in (or ratio of) restricted mean survival time, and the weighted log-rank test. Despite the multitude of options, the convention in survival studies is to assume proportional hazards and to use the unweighted log-rank test for design and analysis. This package provides sample size and power calculation for all of the above statistical tests with allowance for flexible accrual, censoring, and survival (eg. Weibull, piecewise-exponential, mixture cure). It is the companion R package to the paper by Yung and Liu (2020) <doi:10.1111/biom.13196>. Specific to the weighted log-rank test, users may specify which approximations they wish to use to estimate the large-sample mean and variance. The default option has been shown to provide substantial improvement over the conventional sample size and power equations based on Schoenfeld (1981) <doi:10.1093/biomet/68.1.316>.

Depends R (>= 3.4.0)

Imports stats, utils

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.1

Suggests knitr, rmarkdown, dplyr, tidyr, tibble, ggplot2

VignetteBuilder knitr

URL https://github.com/godwinyyung/npsurvSS

BugReports https://github.com/godwinyyung/npsurvSS/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2024-05-08 22:00:02 UTC

Contents

`create_arm` *Create an 'arm' object*

Description

Create an object of class 'arm' by specifying the trial parameters for a single arm, including the sample size, accrual distribution, survival distribution, loss to follow-up distribution, and study duration.

Usage

```
create_arm(
  size,
  accr_time,
  accr_dist = "pieceuni",
  accr_interval = c(0, accr_time),
  accr_param = NA,
  surv_cure = 0,
  surv_interval = c(0, Inf),
  surv_shape = 1,
  surv_scale,
  loss_shape = 1,
  loss_scale,
```

```
  follow_time = Inf,
  total_time = Inf
)
```

Arguments

size	sample size. If total sample size is unknown, provide the integer sample size relative to the opposing arm, e.g. 1 for 1:2 randomization ratio or 2 for 2:3.
accr_time	accrual duration.
accr_dist	accrual distribution. Default is piecewise uniform. Alternatively, 'truncexp' allows for a truncated exponential distribution as proposed by Lachin and Foulkes (1986). Depending on the value of accr_param, this distribution can be either convex or concave.
accr_interval	accrual intervals. Defaults to the single interval spanning from 0 to accr_time. If a piecewise uniform accrual with more than one interval is desired, specify accr_interval as the vector of increasing changepoints (knots) starting from 0 and ending with accr_time, e.g. c(0, 2, 4) defines a piecewise uniform distribution with two intervals, [0, 2) and [2, 4].
accr_param	additional accrual parameter(s). For a piecewise uniform accrual with more than one interval, specify accr_param as the vector of probabilities a patient is enrolled in each interval. The probabilities should naturally sum to 1. For accr_dist='truncexp', specify accr_param as a single number. >0 results in a convex distribution and <0 results in a concave distribution.
surv_cure	proportion of patients that are cured.
surv_interval	survival intervals. Defaults to the single interval spanning from 0 to infinity. If a piecewise exponential survival is desired for uncured patients, specify surv_interval as the vector of increasing changepoints (knots) starting from 0 and ending with infinity, e.g. c(0, 6, 10, Inf).
surv_shape	Weibull shape parameter for the survival distribution of uncured patients.
surv_scale	Weibull scale parameter for the survival distrubition of uncured patients. Piecewise exponential survival may be defined by specifying surv_shape=1 and surv_scale as the vector of piecewise hazard rates.
loss_shape	Weibull shape parameter for the loss to follow-up distribution.
loss_scale	Weibull scale parameter for the loss to follow-up distribution.
follow_time	follow-up duration.
total_time	total study duration. Only 1 of the 2 parameters, follow_time or total_time, need to be defined. If neither is defined, total_time is defaulted to max value 1e6.

Value

a list containing assumptions of size, accrual, censoring, survival, and follow-up for a single arm.

References

Lachin, J. M. and Foulkes, M. A. (1986) Evaluation of sample size and power for analyses of survival with allowance for nonuniform patient entry, losses to follow-up, noncompliance, and stratification. *Biometrics*, **42**, 507-519.

See Also

create_arm_lachin for creating an object of subclass 'lachin'.

Examples

```
# Example 1
example <- create_arm(size=120,
  accr_time=6,                  # uniform accrual
  surv_scale=0.05,              # exponential survival
  loss_scale=0.005,             # exponential loss to follow-up
  follow_time=12)
class(example)              # this example also satisfies properties of subclass 'lachin'

# Example 2
create_arm(size=120,
  accr_time=6,                  # truncated exponential accrual
  accr_dist="truncexp",
  accr_param=0.1,
  surv_shape=2,                 # weibull survival
  surv_scale=0.05,
  loss_shape=1.5,               # weilbull loss to follow-up
  loss_scale=0.005,
  total_time=18)

# Example 3
create_arm(size=120,
  accr_time=6,
  accr_interval=c(0,2,4,6),     # piecewise uniform accrual
  accr_param=c(0.2,0.3,0.5),
  surv_cure=0.1,                # 10% cure fraction
  surv_interval=c(0,6,10,Inf),  # piecewise exponential survival for uncured patients
  surv_scale=c(0.05,0.04,0.03),
  loss_shape=0.7,               # weibull loss to follow-up
  loss_scale=0.005,
  total_time=18)
```

---

create_arm_lachin *Create a 'lachin' object*

Description

Create an object of class 'lachin' by specifying the trial parameters for a single arm, including the sample size, accrual distribution, survival distribution, loss to follow-up distribution, and study duration. 'Lachin' objects are also 'arm' objects, but with accrual limited to the uniform and truncated

exponential distributions, and survival and loss to follow-up limited to the exponential distribution. 'Lachin' objects have the advantage that expectations for certain counting processes have closed form equations and can therefore be calculated more efficiently (Lachin and Foulkes, 1986).

Usage

```
create_arm_lachin(
  size,
  accr_time,
  accr_dist = "pieceuni",
  accr_param = NA,
  surv_median = NA,
  surv_exphazard = NA,
  surv_milestone = NA,
  loss_median = NA,
  loss_exphazard = NA,
  loss_milestone = NA,
  follow_time = Inf,
  total_time = Inf
)
```

Arguments

`size`	sample size. If total sample size is unknown, provide the integer sample size relative to the opposing arm, e.g. 1 for 1:2 randomization ratio or 2 for 2:3.
`accr_time`	accrual duration.
`accr_dist`	accrual distribution. Default is uniform (piecewise uniform with one interval). Alternatively, 'truncexp' allows for a truncated exponential distribution as proposed by Lachin and Foulkes (1986). Depending on the value of `accr_param`, this distribution can be either convex or concave.
`accr_param`	additional accrual parameter for `accr_dist='truncexp'`. `accr_param`>0 specifies a convex distribution and `accr_param`<0 specifies a concave distribution.
`surv_median`	median survival.
`surv_exphazard`	exponential hazard rate for the survival distribution.
`surv_milestone`	a tuple c(milestone, probability) that uniquely defines the exponential survival distribution, e.g. c(12, 0.8) corresponds to the exponential distribution with 80% survival rate at 12 months.
`loss_median`	median loss to follow-up.
`loss_exphazard`	exponential hazard rate for the loss to follow-up distribution.
`loss_milestone`	a tuple c(milestone, probability) that uniquely defines the exponential loss to follow-up distribution, e.g. c(12, 0.99) corresponds to the exponential distribution with 1% loss to follow-up at 12 months.
`follow_time`	Follow-up duration. Either `follow_time` or `total_time` (below) should be specified.
`total_time`	Total study duration. Either `follow_time` (above) or `total_time` should be specified.

Value

a list containing assumptions of size, accrual, censoring, survival, and follow-up for a single arm.

References

Lachin, J. M. and Foulkes, M. A. (1986) Evaluation of sample size and power for analyses of survival with allowance for nonuniform patient entry, losses to follow-up, noncompliance, and stratification. *Biometrics*, **42**, 507-519.

See Also

`create_arm` for creating an object of class 'arm'.

Examples

```
# 3 arms with similar survival and loss to follow-up
create_arm_lachin(size=120, accr_time=6,
  surv_median=14,
  loss_median=140,
  follow_time=12)
create_arm_lachin(size=120, accr_time=6,
  surv_exphazard=0.05,
  loss_exphazard=0.005,
  follow_time=12)
create_arm_lachin(size=120, accr_time=6,
  accr_dist="truncexp",
  accr_param=0.1,
  surv_milestone=c(14, 0.5),
  loss_milestone=c(140, 0.5),
  total_time=18)
```

---

daccr *Accrual*

Description

Density, distribution function, quantile function, and random generation for the accrual distribution.

Usage

```
daccr(x, arm)

paccr(q, arm, lower.tail = T)

qaccr(p, arm)

raccr(n = 1, arm)
```

Arguments

x, q	vector of quantiles.
arm	object of class 'arm'.
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
p	vector of probabilities.
n	number of observations.

Value

daccr gives the density, paccr gives the distribution function, qaccr gives the quantile function, and raccr generates random deviates.

See Also

create_arm and create_arm_lachin for creating an object of class 'arm'.

---

dloss *Loss to follow-up*

Description

Density, distribution function, hazard function, quantile function, and random generation for the loss to follow-up distribution.

Usage

```
dloss(x, arm)

ploss(q, arm, lower.tail = T)

hloss(x, arm)

qloss(p, arm, lower.tail = T)

rloss(n = 1, arm)
```

Arguments

x, q	vector of quantiles.
arm	object of class 'arm'.
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
p	vector of probabilities.
n	number of observations.

Value

dloss gives the density, ploss gives the distribution function, hloss gives the hazard function, qloss gives the quantile function, and rloss generates random deviates.

See Also

create_arm and create_arm_lachin for creating an object of class 'arm'.

---

dmaxU *Maximum observed time*

Description

Density, distribution function, and expected value for the maximum observed time in a single arm of patients.

Usage

```
dmaxU(x, arm, include_cens = T)

pmaxU(q, arm, include_cens = T, lower.tail = T)

emaxU(arm, include_cens = T)
```

Arguments

x, q	vector of quantiles.
arm	object of class 'arm'.
include_cens	logical; if TRUE, include time-to-censoring as potential observed time; otherwise, observed time equals time-to-event.
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.

Details

Given a patient's time-to-event T_i and time-to-censoring C_i, $U_i = \min(T_i, C_i)$ defines the patient's observed time. The maximum observed time over patients of a single arm is then $\max_i U_i$.

Value

dmaxU gives the density, pmaxU gives the distribution function, and emaxU gives the expected value.

See Also

create_arm and create_arm_lachin for creating an object of class 'arm'.

dminimaxU *Minimax observed time*

Description

Density, distribution function, quantile function, and expected value for the minimum of the maximum observed time over two treatment arms.

Usage

```
dminimaxU(x, arm0, arm1, include_cens = T)

pminimaxU(q, arm0, arm1, include_cens = T, lower.tail = T)

qminimaxU(p, arm0, arm1, include_cens = T, margin = 0.01)

eminimaxU(arm0, arm1, include_cens = T)
```

Arguments

x, q	vector of quantiles.
arm0	object of class 'arm'.
arm1	object of class 'arm'.
include_cens	logical; if TRUE, include time-to-censoring as potential observed time; otherwise, observed time equals time-to-event.
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
p	vector of probabilities.
margin	margin of accuracy.

Details

Given a patient in arm $X_i = j$ with time-to-event T_i and time-to-censoring C_i, $U_i = \min(T_i, C_i)$ defines the patient's observed time. The maximum observed time over patients of arm j is then $\max_{i:X_i=j} U_i$, and the minimax observed time over two arms is $\min_j(\max_{i:X_i=j} U_i)$.

Value

dminimaxU gives the density, pminimaxU gives the distribution function, qminimaxU gives the quantile function, and eminimaxU gives the expected value.

See Also

create_arm and create_arm_lachin for creating an object of class 'arm'.

dsurv *Survival*

Description

Density, distribution function, hazard function, quantile function, and random generation for the survival distribution.

Usage

```
dsurv(x, arm, include_cured = T)

psurv(q, arm, include_cured = T, lower.tail = T)

hsurv(x, arm, include_cured = T)

qsurv(p, arm, include_cured = T, lower.tail = T)

rsurv(n = 1, arm, include_cured = T)
```

Arguments

x, q	vector of quantiles.
arm	object of class arm.
include_cured	logical; if TRUE, mixture distribution of cured and uncured patients is considered; otherwise, only the distribution for uncured patients is considered.
lower.tail	logical; if TRUE, probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
p	vector of probabilities.
n	number of observations.

Value

dsurv gives the density, psurv gives the distribution function, hsurv gives the hazard function, qsurv gives the quantile function, and rsurv generates random deviates.

See Also

create_arm and create_arm_lachin for creating an object of class 'arm'.

exp_duration *Expected duration*

Description

Given one or two treatment arms, calculate the time τ at which the expected number of events equals d.

Usage

```
exp_duration(
  arm0 = NA,
  arm1 = NA,
  d,
  search_start = 10,
  search_prec = 0.01,
  max_duration = 1000
)
```

Arguments

arm0	object of class 'arm'.
arm1	object of class 'arm'.
d	vector of number of events.
search_start	value at which the search for duration tau starts.
search_prec	value controlling the desired precision before terminating the search.
max_duration	maximum τ for consideration.

Value

expected trial duration.

See Also

exp_events for calculating expected events d at time tau, create_arm and create_arm_lachin for creating an object of class 'arm'.

Examples

```
arm0 <- create_arm(size=120, accr_time=6, surv_scale=0.05, loss_scale=0.005, follow_time=12)
arm1 <- create_arm(size=120, accr_time=6, surv_scale=0.03, loss_scale=0.005, follow_time=12)
exp_duration(arm0, d=61)
exp_duration(arm0, arm1, d=103)
exp_duration(arm0, d=c(35,61))
exp_duration(arm0, arm1, d=c(57,103))
```

exp_events *Expected number of events*

Description

Given one or two treatment arms, calculate the expected number of events d at time τ.

Usage

```
exp_events(arm0 = NA, arm1 = NA, tau = NA)
```

Arguments

arm0	object of class 'arm'.
arm1	object of class 'arm'.
tau	vector of times. Defaults to total study duration.

Value

expected number of events.

See Also

exp_duration for calculating time to achieve expected events d, create_arm and create_arm_lachin for creating an object of class 'arm'.

Examples

```
arm0 <- create_arm(size=120, accr_time=6, surv_scale=0.05, loss_scale=0.005, follow_time=12)
arm1 <- create_arm(size=120, accr_time=6, surv_scale=0.03, loss_scale=0.005, follow_time=12)
exp_events(arm0)
exp_events(arm0, arm1)
exp_events(arm0, tau=c(10,NA))
exp_events(arm0, arm1, tau=c(10,NA))
```

per2haz *Convert exponential parameters*

Description

Convert exponential survival percentile or hazard rate to the other.

Usage

```
per2haz(x, per = 0.5)
```

Arguments

x	survival percentile or exponential hazard rate
per	(per)th percentile

Details

$$y = -log(1 - per)/x$$

Value

survival percentile or hazard rate.

Examples

```
per2haz(14)          # hazard rate for exponential with 14-month median
per2haz(0.05)        # median survival for exponential with hazard rate 0.05
per2haz(14, 0.8)    # hazard rate for exponential with 80th percentile survival at 14 months
per2haz(0.27, 0.8)   # 80th percentile survival for exponential with hazard rate 0.27
```

---

power_two_arm *Power*

Description

Calculate power for a two-arm survival study.

Usage

```
power_two_arm(
  arm0,
  arm1,
  test = list(test = "weighted logrank"),
  alpha = 0.025,
  sides = 1
)
```

Arguments

arm0	object of class 'arm'.
arm1	object of class 'arm'.
test	list or list of lists. Each list must contain at minimum the key 'test' describing the type of statistical test. Default test is the "weighted logrank". Kaplan-Meier based tests ("survival difference", "survival ratio", "rmst difference", "rmst ratio", "percentile difference", and "percentile ratio") require the user to define an additional key, either the desired 'milestone' or 'percentile'. The weighted log-rank test does not require additional keys. However, user may choose which

	weight function ("1"=unweighted, "n"=Gehan-Breslow, "sqrtN"=Tarone-Ware, "FH_p[a]_q[b]"= Fleming-Harrington with p=a and q=b) and which approximation for the large-sample mean ("asymptotic", "generalized schoenfeld", "event driven", "freedman", "rubinstein") and variance ("1", "block[randomization]", "simple[randomization]") they wish to use. Default choice is 'weight'="1", 'mean.approx'="asymptotic", and 'var.approx'="1". For more details regarding the different mean and variance approximations for the weight log-rank test, please see Yung and Liu (2020). If there are multiple lists, then users may provide a 'label' for each list to be displayed in the output.
alpha	type 1 error rate
sides	1=1-sided test, 2=2-sided test

Value

power.

References

Yung, G and Liu, Y. (2020). Sample size and power for the weighted log-rank test and Kaplan-Meier based tests with allowance for non-proportional hazards. *Biometrics* 76(3):939-950.

See Also

create_arm for creating an object of class 'arm'.

Examples

```
arm0 <- create_arm(size=120, accr_time=6, surv_scale=0.05, loss_scale=0.005, follow_time=12)
arm1 <- create_arm(size=120, accr_time=6, surv_scale=0.03, loss_scale=0.005, follow_time=12)
power_two_arm(arm0, arm1)
power_two_arm(arm0, arm1, list(test="weighted logrank",
  weight="n",
  mean.approx="generalized schoenfeld",
  var.approx="block"))
power_two_arm(arm0, arm1, list(test="survival difference", milestone=12))
power_two_arm(arm0, arm1, list(test="rmst ratio", milestone=12))
power_two_arm(arm0, arm1, list(test="percentile difference", percentile=0.25))
power_two_arm(arm0, arm1, list(
  list(test="weighted logrank", label="Logrank"),
  list(test="survival difference", milestone=12, label="12-month survival difference")))
```

---

simulate_arm *Simulate complete data for a single arm*

Description

Simulate the complete data for a single arm, including time to accrual, event, and loss of follow-up. No cutoff (by number of events or time) is applied. Hence, no patients are administratively censored.

Usage

```
simulate_arm(arm, label = 1)
```

Arguments

arm	object of class 'arm'.
label	numeric label for the simulated arm, e.g. 0 for control, 1 for treatment

Value

arm	label
time.accr	time to accrual
time.obs	time to observation from accrual
time.total	time to observation from start of study
censor	0=censor, 1=event
reason	event description ('[experience]event', '[loss to]followup', 'administration[censoring]')
time.surv	time to event
time.loss	time to loss of follow-up

See Also

create_arm for creating an object of class 'arm'.

Examples

```
arm0 <- create_arm(size=120, accr_time=6, surv_scale=0.05, loss_scale=0.005, follow_time=12)
simulate_arm(arm0, label=0)
```

---

simulate_trial *Simulate a clinical trial*

Description

Simulate a single- or two-arm clinical trial, where end of study (EOS) is triggered after a number of events has been observed or a certain time has elapsed. Whereas simulate_arm provides complete data for patients, including time to event and loss of follow-up, simulate_trial mimicks an actual survival study by providing only the observed time (minimum of time to event or censoring) and censoring indicator.

Usage

```
simulate_trial(arm0 = NA, arm1 = NA, events = NA, duration = Inf)
```

Arguments

arm0	object of class 'arm'.
arm1	object of class 'arm'.
events	number of required events to trigger end of study; overrides study duration defined within arm0 and arm1.
duration	time from first-patient-in to trigger end of study; overrides study duration defined within arm0 and arm1. If both events and duration are specified, end of study is triggered by either criteria, whichever occurs first.

Value

arm	0=arm0, 1=arm1
time.accr	time to accrual
time.obs	time to observation from accrual
time.total	time to observation from start of study
censor	0=censor, 1=event
reason	event description ('[experience]event', '[loss to]followup', 'administration[censoring]')

See Also

simulate_arm for simulating complete data for a single arm, create_arm for creating an object of class 'arm'.

Examples

```
arm0 <- create_arm(size=120, accr_time=6, surv_scale=0.05, loss_scale=0.005, follow_time=12)
arm1 <- create_arm(size=120, accr_time=6, surv_scale=0.03, loss_scale=0.005, follow_time=12)
simulate_trial(arm0, duration=10)
simulate_trial(arm0, arm1, events=50)
```

---

size_two_arm *Sample size*

Description

Calculate required sample size and expected number of events for a two-arm survival study.

Usage

```
size_two_arm(
  arm0,
  arm1,
  test = list(test = "weighted logrank"),
  power = 0.8,
  alpha = 0.025,
  sides = 1
)
```

Arguments

arm0	object of class 'arm'.
arm1	object of class 'arm'.
test	list or list of lists. Each list must contain at minimum the key 'test' describing the type of statistical test. Default test is the "weighted logrank". Kaplan-Meier based tests ("survival difference", "survival ratio", "rmst difference", "rmst ratio", "percentile difference", and "percentile ratio") require the user to define an additional key, either the desired 'milestone' or 'percentile'. The weighted log-rank test does not require additional keys. However, user may choose which weight function ("1"=unweighted, "n"=Gehan-Breslow, "sqrtN"=Tarone-Ware, "FH_[a]_[b]"= Fleming-Harrington with p=a and q=b) and which approximation for the large-sample mean ("asymptotic", "generalized schoenfeld", "event driven", "freedman", "rubinstein") and variance ("1", "block[randomization]", "simple[randomization]") they wish to use. Default choice is 'weight'="1", 'mean.approx'="asymptotic", and 'var.approx'="1". For more details regarding the different mean and variance approximations for the weight log-rank test, please see Yung and Liu (2020). If there are multiple lists, then users may provide a 'label' for each list to be displayed in the output.
power	1 - type 2 error rate
alpha	type 1 error rate
sides	1=1-sided test, 2=2-sided test

Value

n0	sample size for arm0
n1	sample size for arm1
n	total sample size
d0	expected number of events for arm0
d1	expected number of events for arm1
d	total expected number of events; can be used to convert a time-driven trial to an event-driven trial.

References

Yung, G and Liu, Y. (2020). Sample size and power for the weighted log-rank test and Kaplan-Meier based tests with allowance for non-proportional hazards. *Biometrics* 76(3):939-950.

See Also

create_arm for creating an object of class 'arm'.

Examples

```
arm0 <- create_arm(size=120, accr_time=6, surv_scale=0.05, loss_scale=0.005, follow_time=12)
arm1 <- create_arm(size=120, accr_time=6, surv_scale=0.03, loss_scale=0.005, follow_time=12)
size_two_arm(arm0, arm1)
```

```
size_two_arm(arm0, arm1, list(test="weighted logrank",
  weight="n",
  mean.approx="generalized schoenfeld",
  var.approx="block"))
size_two_arm(arm0, arm1, list(test="survival difference", milestone=12))
size_two_arm(arm0, arm1, list(test="rmst ratio", milestone=12))
size_two_arm(arm0, arm1, list(test="percentile difference", percentile=0.25))
size_two_arm(arm0, arm1, list(
  list(test="weighted logrank", label="Logrank"),
  list(test="survival difference", milestone=12, label="12-month survival difference")))
```

Index