Package: nphRCT (via r-universe)

September 26, 2024

Title Non-Proportional Hazards in Randomized Controlled Trials

Version 0.1.1

Description Perform a stratified weighted log-rank test in a randomized controlled trial. Tests can be visualized as a difference in average score on the two treatment arms. These methods are described in Magirr and Burman (2018) <doi:10.48550/arXiv.1807.11097>, Magirr (2020) <doi:10.48550/arXiv.2007.04767>, and Magirr and Jimenez (2022) <doi:10.48550/arXiv.2201.10445>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0), dplyr, cowplot, survminer

VignetteBuilder knitr

Config/testthat/edition 3

Depends R (>= 3.5.0)

Imports ggplot2, purrr, survival

LazyData true

NeedsCompilation no

Author Dominic Magirr [aut, cre], Isobel Barrott [aut], Jose Jimenez [ctb]

Maintainer Dominic Magirr <d.magirr@gmail.com>

Repository CRAN

Date/Publication 2024-06-27 12:30:02 UTC

Contents

`find_at_risk` *Calculate at-risk table*

Description

This function calculates the number of individuals at risk and number of events at each distinct event time (and censoring time if `include_cens==TRUE`).

Usage

```
find_at_risk(formula, data, include_cens = TRUE, timefix = TRUE)
```

Arguments

`formula`	Formula object. The response (on the left of the ~ operator) must be a survival object as returned by the `Surv` function. The terms (on the right of the ~ operator) must include the treatment arm indicator, and additionally can include strata using the `strata` function.
`data`	Data frame containing time-to-event data.
`include_cens`	Boolean indicating whether to include values corresponding to censoring times
`timefix`	Deal with floating point issues (as in the survival package). Default is TRUE. May need to set FALSE for simulated data.

Value

Data frame

Returns a data frame with the following columns:

- time `t_j`
- number of events in each of the treatments at `t_j`
- combined number of events in both treatments at event time `t_j`
- number of individuals at risk in each of the treatment groups just before time `t_j`
- combined number of individuals at risk in both treatment groups just before time `t_j`

Examples

```
library(nphRCT)
set.seed(1)
sim_data <- sim_events_delay(
  event_model=list(
    duration_c = 36,
    duration_e = c(6,30),
    lambda_c = log(2)/9,
    lambda_e = c(log(2)/9,log(2)/18)
  ),
  recruitment_model=list(
    rec_model="power",
    rec_period = 12,
    rec_power = 1
  ),
  n_c=5,
  n_e=5,
  max_cal_t = 36
)
#with censoring times included
find_at_risk(formula=survival::Surv(event_time,event_status)~group,
  data=sim_data,
  include_cens=TRUE)
#with censoring times excluded
find_at_risk(formula=survival::Surv(event_time,event_status)~group,
  data=sim_data,
  include_cens=FALSE)
```

`find_scores` *Calculate scores*

Description

Weighted log-rank tests can also be thought in terms of assigning a score to the each of the events (including censoring) and comparing the average score on each arm, see Magirr (2021) doi: 10.1002/pst.2091. This function calculates the scores for different types of weighted log-rank test, the modestly-weighted log-rank test and the Fleming-Harrington (ρ,γ) test, in addition to the standard log-rank test.

Usage

```
find_scores(
  formula,
  data,
  method,
  t_star = NULL,
  s_star = NULL,
  rho = NULL,
  gamma = NULL,
```

```
  tau = NULL,
  timefix = TRUE
)
```

Arguments

formula	Formula object. The response (on the left of the ~ operator) must be a survival object as returned by the Surv function. The terms (on the right of the ~ operator) must include the treatment arm indicator, and additionally can include strata using the strata function.
data	Data frame containing time-to-event data.
method	Character string specifying type of method to calculate scores. Either one of the weighted log-rank tests (log-rank "lr", Fleming-Harrington "fh", modestly weighted "mw") or pseudovalue-based scores (restricted mean survival time "rmst", milestone analysis "ms")
t_star	Parameter t^* in the modestly weighted ("mw") test, see Details.
s_star	Parameter s^* in the modestly weighted ("mw") test, see Details.
rho	Parameter ρ in the Fleming-Harrington ("fh") test, see Details.
gamma	Parameter γ in the Fleming-Harrington ("fh") test, see Details.
tau	Parameter τ in the RMST ("rmst") or milestone analysis ("ms") test.
timefix	Deal with floating point issues (as in the survival package). Default is TRUE. May need to set FALSE for simulated data.

Details

Select which of the tests to perform using argument method. For the weighted log-rank tests, the output is calculated as outlined in vignette("weighted_log_rank_tests", package="nphRCT").

Value

Data frame. Each row corresponds to an event or censoring time. At each time specified in t_j the columns indicate

- event the event indicator
- group the treatment arm indicator
- score the assigned score at time t_j
- standardized_score the value of score standardized to be between -1 and 1

References

Magirr, D. (2021). Non-proportional hazards in immuno-oncology: Is an old perspective needed?. Pharmaceutical Statistics, 20(3), 512-527. doi:10.1002/pst.2091

Magirr, D. and Burman, C.F., 2019. Modestly weighted logrank tests. Statistics in medicine, 38(20), 3782-3790.

Examples

```
library(nphRCT)
set.seed(1)
sim_data <- sim_events_delay(
  event_model=list(
    duration_c = 36,
    duration_e = c(6,30),
    lambda_c = log(2)/9,
    lambda_e = c(log(2)/9,log(2)/18)
  ),
  recruitment_model=list(
    rec_model="power",
    rec_period = 12,
    rec_power = 1
  ),
  n_c=50,
  n_e=50,
  max_cal_t = 36
)
df_scores<-find_scores(formula=Surv(event_time,event_status)~group,
  data=sim_data,
  method="mw",
  t_star = 4
)
plot(df_scores)
```

`find_weights` *Calculate weights*

Description

This function can perform two types of weighted log-rank test, the modestly-weighted log-rank test and the Fleming-Harrington (ρ,γ) test, in addition to the standard log-rank test.

Usage

```
find_weights(
  formula,
  data,
  method,
  t_star = NULL,
  s_star = NULL,
  rho = NULL,
  gamma = NULL,
  include_cens = FALSE,
  timefix = TRUE
)
```

Arguments

- `formula` Formula object. The response (on the left of the ~ operator) must be a survival object as returned by the `Surv` function. The terms (on the right of the ~ operator) must include the treatment arm indicator, and additionally can include strata using the `strata` function.
- `data` Data frame containing time-to-event data.
- `method` Character string specifying type of weighted log-rank test. Either `"lr"` for a standard log-rank test, `"mw"` for a modestly-weighted log-rank test, or `"fh"` for the Fleming-Harrington rho-gamma family.
- `t_star` Parameter t^* in the modestly weighted (`"mw"`) test, see Details.
- `s_star` Parameter s^* in the modestly weighted (`"mw"`) test, see Details.
- `rho` Parameter ρ in the Fleming-Harrington (`"fh"`) test, see Details.
- `gamma` Parameter γ in the Fleming-Harrington (`"fh"`) test, see Details.
- `include_cens` Boolean indicating whether to include values corresponding to censoring times
- `timefix` Deal with floating point issues (as in the survival package). Default is TRUE. May need to set FALSE for simulated data.

Details

Select which of the three tests to perform using argument `method`. The output is calculated as outlined in `vignette("weighted_log_rank_tests", package="nphRCT")`.

Value

Vector of weights in the weighted log-rank test. The weights correspond to the ordered, distinct event times (and censoring times if `include_cens=TRUE`).

References

Magirr, D. (2021). Non-proportional hazards in immuno-oncology: Is an old perspective needed?. Pharmaceutical Statistics, 20(3), 512-527. doi:10.1002/pst.2091

Magirr, D. and Burman, C.F., 2019. Modestly weighted logrank tests. Statistics in medicine, 38(20), 3782-3790.

Examples

```
library(nphRCT)
set.seed(1)
sim_data <- sim_events_delay(
  event_model=list(
    duration_c = 36,
    duration_e = c(6,30),
    lambda_c = log(2)/9,
    lambda_e = c(log(2)/9,log(2)/18)
  ),
  recruitment_model=list(
    rec_model="power",
```

```
    rec_period = 12,
    rec_power = 1
  ),
  n_c=5,
  n_e=5,
  max_cal_t = 36
)
#example setting t_star
find_weights(formula=Surv(event_time,event_status)~group,
  data=sim_data,
  method="mw",
  t_star = 4
)
```

`moderate_cross` *Time-to-event RCT data set with moderate crossing of survival curves.*

Description

A synthetic data set based on an RCT with crossing survival curves.

Usage

```
moderate_cross
```

Format

A data frame with 328 rows and 3 variables:

time time to event / censoring

event observed event 1 / 0

arm treatment arm ...

`sim_events_delay` *Simulate survival data from a two-arm trial*

Description

Simulate survival data from a two-arm trial with survival times on the control arm and experimental arm simulated from an exponential distribution or piecewise exponential distribution.

Usage

```
sim_events_delay(event_model, recruitment_model, n_c, n_e, max_cal_t)
```

Arguments

event_model — List containing information to simulate event times, including:

- duration_c Vector of durations corresponding to each of the periods of the control arm.
- duration_e Vector of durations corresponding to each of the periods of the experimental arm.
- lambda_c Vector of parameters λ in the exponential distribution corresponding to each of the periods of the control arm.
- lambda_e Vector of parameters λ in the exponential distribution corresponding to each of the periods of the experimental arm.

recruitment_model — List containing information to simulate recruitment times, including:

- rec_model Character string specifying the type of recruitment model. Either the power model "power" or piecewise constant model "pw_constant".
- rec_power Parameter used to model recruitment according to power model, see Details.
- rec_period Parameter used to model recruitment according to power model, see Details.
- rec_rate Parameter used to model recruitment according to piecewise constant model, see Details.
- rec_duration Parameter used to model recruitment according to piecewise constant model, see Details.

n_c — Number of individuals on the control arm

n_e — Number of individuals on the event arm

max_cal_t — Calendar time at which the trial ends, all observations are censored at this time.

Details

Survival times are simulated from an exponential distribution with rate parameter λ, $f(t) = \lambda exp(-\lambda t)$. This distribution has a median value of $log(2)/\lambda$; this can be a useful fact when setting the rates lambda_c and lambda_e. The survival times can be simulated from a piecewise exponential distribution, setting one/multiple durations and λ parameters for the control and experimental arms.

Recruitment is modeled using either the power model or the piecewise constant model.

The power model is defined as: $P(recruited\_before\_T) = (T/rec\_period)^{rec\_power}$, where rec_period is the time at the end of recruitment period, and rec_power controls the rate of recruitment.

Alternatively, recruitment can be modelled using the piecewise constant model. In the simple case with only one time period defined in rec_duration, the times between each of the individuals entering follow-up are samples from the exponential distribution with rate parameter λ, $f(t) = \lambda exp(-\lambda t)$. The number of recruitment times defined in n_c or n_e is returned, regardless of the length of duration rec_duration.

In the case with multiple time periods defined in rec_duration, the number of events in each period is sampled from the Poisson distribution $P(K = k) = \lambda^k \exp(-\lambda/k!)$, where k is the number of events. The rate parameter λ is equal to rec_rate multiplied by the duration of the time period in rec_duration. The recruitment times are then sampled uniformly from the corresponding time period. In the case that insufficient recruitment times have been simulated by the end of the last time period, the additional recruitment times will be simulated after the end of the last time period.

All observations are censored at the calendar time defined in argument max_cal_t.

Value

Data frame with columns event_time, event_status (1 = event, 0 = censored), and treatment arm indicator group.

Examples

```
library(nphRCT)
set.seed(1)
sim_data <- sim_events_delay(
  event_model=list(
    duration_c = 36,
    duration_e = c(6,30),
    lambda_c = log(2)/9,
    lambda_e = c(log(2)/9,log(2)/18)
  ),
  recruitment_model=list(
    rec_model="power",
    rec_period = 12,
    rec_power = 1
  ),
  n_c=50,
  n_e=50,
  max_cal_t = 36
)
```

---

wlrt *Weighted log-rank test*

Description

This function can perform two types of weighted log-rank test, the modestly-weighted log-rank test and the Fleming-Harrington (ρ,γ) test, in addition to the standard log-rank test.

Usage

```
wlrt(
  formula,
  data,
  method,
  t_star = NULL,
  s_star = NULL,
  rho = NULL,
```

```
  gamma = NULL,
  timefix = TRUE
)
```

Arguments

formula	Formula object. The response (on the left of the ~ operator) must be a survival object as returned by the Surv function. The terms (on the right of the ~ operator) must include the treatment arm indicator, and additionally can include strata using the strata function.
data	Data frame containing time-to-event data.
method	Character string specifying type of weighted log-rank test. Either "lr" for a standard log-rank test, "mw" for a modestly-weighted log-rank test, or "fh" for the Fleming-Harrington rho-gamma family.
t_star	Parameter t^* in the modestly weighted ("mw") test, see Details.
s_star	Parameter s^* in the modestly weighted ("mw") test, see Details.
rho	Parameter ρ in the Fleming-Harrington ("fh") test, see Details.
gamma	Parameter γ in the Fleming-Harrington ("fh") test, see Details.
timefix	Deal with floating point issues (as in the survival package). Default is TRUE. May need to set FALSE for simulated data.

Details

Select which of the three tests to perform using argument method. The output is calculated as outlined in vignette("weighted_log_rank_tests", package="wlrt").

Value

List containing the outcome of the weighted log-rank test.

- u is the test statistic U for the weighted log-rank test
- v_u is the variance of test statistic U
- z is the Z-score
- trt_group indicates which of the treatment arms the test statistic U corresponds to

In the presence of multiple strata, the results of the test on each individual strata is returned, in addition to the combined test that was proposed by Magirr and Jiménez (2022), see vignette("weighted_log_rank_tests", package="wlrt").

References

Magirr, D. (2021). Non-proportional hazards in immuno-oncology: Is an old perspective needed?. Pharmaceutical Statistics, 20(3), 512-527. doi:10.1002/pst.2091

Magirr, D. and Burman, C.F., 2019. Modestly weighted logrank tests. Statistics in medicine, 38(20), 3782-3790.

Magirr, D. and Jiménez, J. (2022) Stratified modestly-weighted log-rank tests in settings with an anticipated delayed separation of survival curves PREPRINT at https://arxiv.org/abs/2201.10445

Examples

```
library(nphRCT)
set.seed(1)
sim_data <- sim_events_delay(
  event_model=list(
    duration_c = 36,
    duration_e = c(6,30),
    lambda_c = log(2)/9,
    lambda_e = c(log(2)/9,log(2)/18)
  ),
  recruitment_model=list(
    rec_model="power",
    rec_period = 12,
    rec_power = 1
  ),
  n_c=50,
  n_e=50,
  max_cal_t = 36
)
#example setting t_star
wlrt(formula=Surv(event_time,event_status)~group,
  data=sim_data,
  method="mw",
  t_star = 4
)
#example setting s_star
wlrt(formula=Surv(event_time,event_status)~group,
  data=sim_data,
  method="mw",
  s_star = 0.5
)
#example with 1 strata
sim_data_0 <- sim_data
sim_data_0$ecog=0
sim_data_1 <- sim_events_delay(
  event_model=list(
    duration_c = 36,
    duration_e = c(6,30),
    lambda_c = log(2)/6,
    lambda_e = c(log(2)/6,log(2)/12)
  ),
  recruitment_model=list(
    rec_model="power",
    rec_period = 12,
    rec_power = 1
  ),
  n_c=50,
  n_e=50,
  max_cal_t = 36
)
sim_data_1$ecog=1
sim_data_strata<-rbind(sim_data_0,sim_data_1)
```

```
wlrt(formula=Surv(event_time,event_status)~group+strata(ecog),
  data=sim_data_strata,
  method="mw",
  t_star = 4
)
#example with 2 strata
sim_data_strata_2<-cbind(sim_data_strata,sex=rep(c("M","F"),times=100))
wlrt(formula=Surv(event_time,event_status)~group+strata(ecog)+strata(sex),
  data=sim_data_strata_2,
  method="mw",
  t_star = 4
)
```

Index