Package ‘gscounts’

April 2, 2019

Type Package
Title Group Sequential Designs with Negative Binomial Outcomes
Version 0.1-3
Maintainer Tobias Mütze <tobias.muetze@outlook.com>
Description Design and analysis of group sequential designs for negative
Depends R (>= 3.0.0)
Imports stats, Rcpp(>= 0.12.9)
Suggests testthat, MASS, knitr, rmarkdown, dplyr, gsDesign, mvtnorm
License GPL (>= 2)
NeedsCompilation yes
URL https://github.com/tobiasmuetze/gscounts
BugReports https://github.com/tobiasmuetze/gscounts/issues
VignetteBuilder knitr
LazyData true
LinkingTo Rcpp
Encoding UTF-8
RoxygenNote 6.1.1
Author Tobias Mütze [aut, cre] (<https://orcid.org/0000-0002-4111-1941>)
Repository CRAN
Date/Publication 2019-04-02 17:00:03 UTC

R topics documented:

design_gsnb ................................................. 2
design_nb .................................................. 5
get_calendartime_gsnb .................................... 7
get_info_gsnb .............................................. 8
design_gsnb

Group sequential design with negative binomial outcomes

Description

Design a group sequential trial with negative binomial outcomes

Usage

```r
design_gsnb(rate1, rate2, dispersion, ratio_H0 = 1, random_ratio = 1, 
power, sig_level, timing, esf = obs, esf_futility = NULL, 
futility = NULL, t_recruit1 = NULL, t_recruit2 = NULL, 
study_period = NULL, accrual_period = NULL, followup_max = NULL, 
accrual_speed = 1, ...)
```

Arguments

- `rate1`: numeric; assumed rate of treatment group 1 in the alternative
- `rate2`: numeric; assumed rate of treatment group 2 in the alternative
- `dispersion`: numeric; dispersion (shape) parameter of negative binomial distribution
- `ratio_H0`: numeric; positive number denoting the rate ratio $\mu_1/\mu_2$ under the null hypothesis, i.e. the non-inferiority or superiority margin
- `random_ratio`: numeric; randomization ratio $n1/n2$
- `power`: numeric; target power of group sequential design
- `sig_level`: numeric; Type I error / significance level
- `timing`: numeric vector; 0 < timing[1] < ... < timing[K] = 1 with K the number of analyses, i.e. (K-1) interim analyses and final analysis. When the timing of efficacy and futility analyses differ, timing should not be defined. Instead, the arguments timing_eff and timing_fut have to be used to specify the timing of the efficacy and futility analyses, respectively.
- `esf`: function; error spending function
- `esf_futility`: function; futility error spending function
- `futility`: character; either “binding”, “nonbinding”, or NULL for binding, nonbinding, or no futility boundaries
- `t_recruit1`: numeric vector; recruit (i.e. study entry) times in group 1
$$H_0 : \frac{\mu_1}{\mu_2} \geq \delta \ vs \ H_1 : \frac{\mu_1}{\mu_2} < \delta,$$

with $\delta = \text{ratio}_H$. Non-inferiority of treatment group 1 compared to treatment group 2 is tested for $\delta \in (1, \infty)$. Superiority of treatment group 1 over treatment group 2 is tested for $\delta \in (0, 1]$. The calculation of the efficacy and (non-)binding futility boundaries are performed under the hypothesis $H_0 : \frac{\mu_1}{\mu_2} = \delta$ and under the alternative $H_1 : \frac{\mu_1}{\mu_2} = \frac{\text{rate}_1}{\text{rate}_2}$.

The argument 'accrual_speed' is used to adjust the accrual speed. Number of subjects in the study at study time $t$ is given by $f(t) = a \times t^b$ with $a = n/\text{accrual_period}$ and $b = \text{accrual_speed}$. For linear recruitment, $b = 1$. $b > 1$ results is slower than linear recruitment for $t < \text{accrual_period}$ and faster than linear recruitment for $t > \text{accrual_period}$. Vice verse for $b < 1$.

Value

A list with class "gsnb" containing the following components:

- rate1 as input
- rate2 as input
- dispersion as input
- power as input
- timing as input
- ratio_H0 as input
- ratio_H1 as input
- sig_level as input
- random_ratio as input
- power_fix power of fixed design
- expected_info list; expected information under ratio_H0 and ratio_H1
- efficacy list; contains the elements esf (type I error spending function), spend (type I error spend at each look), and critical (critical value for efficacy testing)
**futility** list; only part of the output if argument futility is defined in the input. Contains the elements futility (input argument futility), esf (type II error spending function), spend (type II error spend at each look), and critical (critical value for futility testing)

**stop_prob** list; contains the element efficacy with the probabilities for stopping for efficacy and, if futility bounds are calculated, the element futility with the probabilities for stopping for futility

**t_recruit1** as input

**t_recruit2** as input

**study_period** as input

**followup_max** as input

**max_info** maximum information

**calendar** calendar times of data looks; only calculated when exposure times are not identical

**References**


**Examples**

# Calculate the sample sizes for a given accrual period and study period (without futility)
out <- design_gsnb(rate1 = 0.0875, rate2 = 0.125, dispersion = 5, 
    power = 0.8, timing = c(0.5, 1), esf = obrien, 
    ratio_H0 = 1, sig_level = 0.025, 
    study_period = 3.5, accrual_period = 1.25, random_ratio = 1)

out

# Calculate the sample sizes for a given accrual period and study period with binding futility
out <- design_gsnb(rate1 = 0.0875, rate2 = 0.125, dispersion = 5, 
    power = 0.8, timing = c(0.5, 1), esf = obrien, 
    ratio_H0 = 1, sig_level = 0.025, study_period = 3.5, 
    accrual_period = 1.25, random_ratio = 1, futility = "binding", 
    esf_futility = obrien)

out

# Calculate study period for given recruitment times
expose <- seq(0, 1.25, length.out = 1042)
out <- design_gsnb(rate1 = 0.0875, rate2 = 0.125, dispersion = 5, 
    power = 0.8, timing = c(0.5, 1), esf = obrien, 
    ratio_H0 = 1, sig_level = 0.025, t_recruit1 = expose, 
    t_recruit2 = expose, random_ratio = 1)

out

# Calculate sample size for a fixed exposure time
out <- design_gsnb(rate1 = 0.0875, rate2 = 0.125, dispersion = 5,
```r
# Different timing for efficacy and futility analyses
design_nb(rate1 = 1, rate2 = 2, dispersion = 5,
    power = 0.8, esf = obrien,
    ratio_H0 = 1, sig_level = 0.025,
    followup_max = 0.5, random_ratio = 1)
```

**Description**

Design a clinical trial with negative binomial outcomes

**Usage**

```r
design_nb(rate1, rate2, dispersion, power, ratio_H0 = 1, sig_level,
    random_ratio = 1, t_recruit1 = NULL, t_recruit2 = NULL,
    study_period = NULL, accrual_period = NULL, followup_max = NULL,
    accrual_speed = 1)
```

**Arguments**

- **rate1** numeric; assumed rate of treatment group 1 in the alternative
- **rate2** numeric; assumed rate of treatment group 2 in the alternative
- **dispersion** numeric; dispersion (shape) parameter of negative binomial distribution
- **power** numeric; target power
- **ratio_H0** numeric; positive number denoting the rate ratio rate_1/rate_2 under the null hypothesis, i.e. the non-inferiority or superiority margin
- **sig_level** numeric; Type I error / significance level
- **random_ratio** numeric; randomization ratio n1/n2
- **t_recruit1** numeric vector; recruit (i.e. study entry) times in group 1
- **t_recruit2** numeric vector; recruit (i.e. study entry) times in group 2
- **study_period** numeric; study duration
- **accrual_period** numeric; accrual period
- **followup_max** numeric; maximum exposure time of a patient
- **accrual_speed** numeric; determines accrual speed; values larger than 1 result in accrual slower than linear; values between 0 and 1 result in accrual faster than linear.
Value

A list containing the following components:

- `rate1` as input
- `rate2` as input
- `dispersion` as input
- `power` as input
- `ratio_H0` as input
- `ratio_H1` ratio rate1/rate2
- `sig_level` as input
- `random_ratio` as input
- `t_recruit1` as input
- `t_recruit2` as input
- `study_period` as input
- `followup_max` as input
- `max_info` maximum information

Examples

# Calculate sample size for given accrual period and study duration assuming uniform accrual
out <- design_nb(rate1 = 0.0875, rate2 = 0.125, dispersion = 5, power = 0.8,
                 ratio_H0 = 1, sig_level = 0.025,
                 study_period = 4, accrual_period = 1, random_ratio = 2)
out

# Calculate sample size for a fixed exposure time of 0.5 years
out <- design_nb(rate1 = 4.2, rate2 = 8.4, dispersion = 3, power = 0.8,
                 ratio_H0 = 1, sig_level = 0.025,
                 followup_max = 0.5, random_ratio = 2)
out

# Calculate study period for given recruitment time
`t_recruit1` <- seq(0, 1.25, length.out = 1200)
`t_recruit2` <- seq(0, 1.25, length.out = 800)
out <- design_nb(rate1 = 0.0875, rate2 = 0.125, dispersion = 5, power = 0.8,
                 ratio_H0 = 1, sig_level = 0.025,
                 t_recruit1 = t_recruit1, t_recruit2 = t_recruit2)
get_calendartime_gsnb  

Calendar time of data looks

Description

Calculate the calendar time of looks given the information time

Usage

get_calendartime_gsnb(rate1, rate2, dispersion, t_recruit1, t_recruit2, timing, followup1, followup2)

Arguments

rate1  numeric; rate in treatment group 1
rate2  numeric; rate in treatment group 2
dispersion  numeric; dispersion (shape) parameter of negative binomial distribution
t_recruit1  numeric vector; recruit (i.e. study entry) times in group 1
t_recruit2  numeric vector; recruit (i.e. study entry) times in group 2
timing  numeric vector with entries in (0,1]; information times of data looks
followup1  numeric vector; final individual follow-up times in treatment group 1
followup2  numeric vector; final individual follow-up times in treatment group 2

Value

numeric; vector with calendar time of data looks

Examples

# Calendar time at which 50%, 75%, and 100% of the maximum information is attained
# 100 subjects in each group are recruited uniformly over 1.5 years
# Study ends after two years, i.e. follow-up times vary between 2 and 0.5 years
get_calendartime_gsnb(rate1 = 0.1, rate2 = 0.125, dispersion = 5, t_recruit1 = seq(0, 1.5, length.out = 100), t_recruit2 = seq(0, 1.5, length.out = 100), timing = c(0.5, 0.75, 1), followup1 = seq(2, 0.5, length.out = 100), followup2 = seq(2, 0.5, length.out = 100))
get_info_gsnb  
*Information level for log rate ratio*

**Description**
Calculates the information level for the log rate ratio of the negative binomial model.

**Usage**
```
get_info_gsnb(rate1, rate2, dispersion, followup1, followup2)
```

**Arguments**
- `rate1`: numeric; rate in treatment group 1
- `rate2`: numeric; rate in treatment group 2
- `dispersion`: numeric; dispersion (shape) parameter of negative binomial distribution
- `followup1`: numeric vector; individual follow-up times in treatment group 1
- `followup2`: numeric vector; individual follow-up times in treatment group 2

**Value**
numeric; information level

**Examples**
```
# Calculates information level for case of 10 subjects per group
# Follow-up times of subjects in each group range from 1 to 3
get_info_gsnb(rate1 = 0.1,
              rate2 = 0.125,
              dispersion = 4,
              followup1 = seq(1, 3, length.out = 10),
              followup2 = seq(1, 3, length.out = 10))
```
**hospitalizations**

**Hospitalizations**

**Description**

A dataset containing the hospitalization times of 1980 patients:

**Usage**

```r
data(hospitalizations)
```

**Format**

A data frame with 2323 rows and 4 variables

**Details**

- treatment. Treatment identifier.
- pat. Patient identifier. Unique within treatment
- t_recruit. Recruitment time of patient into the clinical trial.
- eventtime. Event time of hospitalization. NA corresponds to no event.

---

**obrien**

**obrien**

**Description**

Error spending function mimicking O'Brien & Fleming critical values

**Usage**

```r
obrien(t, sig_level, ...)
```

**Arguments**

- `t` numeric; Non-negative information ratio
- `sig_level` numeric; significance level
- `...` optional arguments

**Value**

numeric

**Examples**

```r
# O'Brien-Fleming-type error spending function
obrien(t = c(0.5, 1), sig_level = 0.025)
```
pocock

Description
Error spending function mimicking Pococks critical values

Usage
pocock(t, sig_level, ...)

Arguments
t numeric; Non-negative information ratio
sig_level numeric; significance level
... optional arguments

Value
numeric

Examples
# Pocock-type error spending function
pocock(t = c(0.5, 1), sig_level = 0.025)

print.gsnb

Description
print method for instance of class gsnb

Usage
## S3 method for class 'gsnb'
print(x, ...)

Arguments
x an object of class gsnb
... optional arguments to print or plot methods
print.nb

Description

print method for instance of class nb

Usage

```r
## S3 method for class 'nb'
print(x, ...)
```

Arguments

- `x`: an object of class `nb`
- `...`: optional arguments to print or plot methods
Index

*Topic datasets
  hospitalizations, 9

design_gsnb, 2
design_nb, 5

g_get_calendartime_gsnb, 7
g_get_info_gsnb, 8
gscounts, 8
gscounts-package (gscounts), 8
hospitalizations, 9
obrien, 9
pocock, 10
print_gsnb, 10
print_nb, 11