Package 'OptOTrials'

August 31, 2026

Type Package

Title Optimal Two-Stage Designs for Ordered Categorical Outcomes

Version 1.0.3

Date 2026-08-26

Description Functions to design and simulate optimal two-stage randomized controlled trials (RCTs) with ordered categorical outcomes, supporting rank-based tests and group-sequential decision rules. Methods build on classical and modern rank tests and two-stage/Group-Sequential designs, e.g., Park (2025) <doi:10.1371/journal.pone.0318211>. The functions 'rule()', 'op()' and 'design_table()' provide a single entry point for constructing designs, evaluating their operating characteristics, and tabulating several designs at once. The earlier functions, one for each combination of test statistic and stopping rule, are retained and still return the same values, but they are deprecated: each warns and names its replacement, and they will be removed in the next version. Please see the package reference manual and the vignette for details.

License GPL-3

Encoding UTF-8

Depends R (>= 4.0)

Imports stats

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Yeonhee Park [aut, cre],
Yudi Wang [aut],
Zhanpeng Xu [aut]

Maintainer Yeonhee Park <yeonheepark@skku.edu>

Repository CRAN

Date/Publication 2026-08-31 18:20:02 UTC

Contents

`OptOTrials-package` *Optimal Two-Stage Designs for Ordered Categorical Outcomes*

Description

Functions to design and simulate optimal two-stage randomized controlled trials (RCTs) with ordered categorical outcomes, supporting rank-based tests and group-sequential decision rules. Methods build on classical and modern rank tests and two-stage/Group-Sequential designs, e.g., Park (2025) <doi: 10.1371/journal.pone.0318211>. The functions 'rule()', 'op()' and 'design_table()' provide a single entry point for constructing designs, evaluating their operating characteristics, and tabulating several designs at once. The earlier functions, one for each combination of test statistic and stopping rule, are retained and still return the same values, but they are deprecated: each warns and names its replacement, and they will be removed in the next version. Please see the package reference manual and the vignette for details.

Details

There are several main functions. `Decision_rule_S_1stage`, `Decision_rule_M_1stage`, `Decision_rule_W_1stage`, `ruleF`, and `ruleFS` determine the decision rule for clinical trials. `op.1stage`, `op.F`, and `op.FS` calculate the operating characteristics for clinical trial designs, including type I error, power, and expected sample size, to investigate the performance of the designs.

Author(s)

Yeonhee Park [aut, cre], Yudi Wang [aut], Zhanpeng Xu [aut]

Maintainer: Yeonhee Park <yeonheepark@skku.edu>

References

Park, Y. (2025). Optimal two-stage group sequential designs based on Mann-Whitney-Wilcoxon test. PloS one, 20(2), e0318211.

`design_table` *Table of designs and operating characteristics*

Description

Builds a data frame of decision rules and simulated operating characteristics over a set of tests and optimality criteria, replacing the `expand.grid()` and `apply()` idiom that required positional indexing of returned vectors. The random number generator is re-seeded for every row, so each row is reproducible on its own.

Usage

```
design_table(alpha, beta, p1, p2, tests, criteria,
             stopping = c("F", "FS"), lambda = 1, nsim = 10000,
            seed = 1234, on_degenerate = c("fallback", "warn", "error", "none"),
             digits = NULL)
```

Arguments

`alpha`, `beta`, `p1`, `p2`, `lambda`	As in `rule`.
`tests`	Character vector of tests, a subset of `c("S", "M", "W")`.
`criteria`	Numeric vector of optimality criteria, a subset of 1 to 5.
`stopping`	`"F"` or `"FS"`.
`nsim`	Number of simulation replicates per row.
`seed`	Seed applied before each scenario of each row.
`on_degenerate`	Passed to `rule`, with the same default, `"fallback"`: a degenerate optimum is replaced by the corresponding single-stage design and the substitution is reported. Set to `"none"` to report the raw optimum instead, in which case the `degenerate` column flags the affected rows and their operating characteristics are returned as `NA`, because a stage-1 total below two patients cannot be allocated to two arms and so cannot be simulated. That is the only case in which the operating characteristics are omitted; any other error from `op` is passed on.
`digits`	Optional number of digits for rounding the estimated probabilities. When supplied, EN0 and ENa are rounded to two decimals and EN is recomputed from the rounded values, so that a reader checking EN = (EN0 + ENa)/2 against the printed table reproduces the printed EN exactly. The default `NULL` returns full precision.

Value

A data frame with one row per (test, criterion) combination, containing the decision rule, the estimated type I error rate and power with Monte Carlo standard errors, the futility and superiority stopping probabilities under each hypothesis, the expected sample sizes, and a `degenerate` flag.

See Also

rule, op

Examples

```
p1 <- c(1/3, 1/3, 1/3)
p2 <- c(1/2, 1/3, 1/6)
design_table(0.05, 0.2, p1, p2, tests = "M", criteria = 1,
             stopping = "FS", nsim = 500)
```

---

op *Operating characteristics of a design*

Description

Estimates the operating characteristics of a design produced by rule by Monte Carlo simulation. Both hypotheses are evaluated in a single call, and every returned quantity is named, so the meaning of the output does not depend on how the function was called.

Usage

```
op(design, nsim = 10000, seed = NULL, p1 = NULL, p2 = NULL,
   allow_degenerate = FALSE)
```

Arguments

design	An object returned by rule.
nsim	Number of simulation replicates (default 10000).
seed	Optional integer. If supplied, the random number generator is seeded immediately before each scenario, so the result is reproducible independently of the state of the stream when op() is reached.
p1, p2	Optional category probabilities overriding those stored in design, for evaluating a design under a scenario other than the one it was built for.
allow_degenerate	Logical. A degenerate two-stage optimum that has not been replaced by its single-stage substitute cannot be simulated meaningfully, because its stage-1 size is too small to allocate patients to both arms; such a design is refused by default. Set to TRUE to override the check.

Value

An object of class "OptOTrialsOC" containing alpha and power with their Monte Carlo standard errors se_alpha and se_power, the expected sample sizes EN0, ENa and EN, and a data frame table reporting, under each hypothesis, the rejection probability, the probability of stopping early for futility, the probability of stopping early for superiority, the probability of continuing to the second stage, and the expected sample size. Futility and superiority stopping are reported separately rather than pooled.

See Also

rule, design_table

Examples

```
p1 <- c(0.075, 0.182, 0.319, 0.243, 0.015, 0.166)
p2 <- p2_fun(p1, log(3.06))
d <- rule(0.05, 0.2, p1, p2, test = "M", stopping = "F", criterion = 1)
op(d, nsim = 1000, seed = 1234)
```

---

p2_fun *p2*

Description

This is the function to calculate the probability p2 when p1 and odds ratio are given.

Usage

```
p2_fun(p1, theta)
```

Arguments

p1	A vector containing the probabilities of the outcome falling into each level of the control arm.
theta	The log odds ratio according to expected effect of the experimental treatment.

Value

A numeric vector representing the expected probability distribution of outcomes across levels for the experimental group.

Proportional_odds_assumption *Checking proportional odds assumption*

Description

This is the function to check the proportional odds assumption for the score test.

Usage

```
Proportional_odds_assumption(p1, p2, verbose = TRUE)
```

Arguments

p1	A vector containing the probabilities of the outcome falling into each level of the control arm.
p2	A vector containing the probabilities of the outcome falling into each level of the experimental arm.
verbose	Logical. If TRUE (the default) an explanatory message is printed stating whether the assumption holds and, if so, the implied common odds ratio.

Value

Indicates whether the proportional odds assumption holds. If the assumption holds, the function returns the log-odds ratio from the score test. If the assumption does not hold, the function returns NA and additionally emits a message and a warning explaining that the score test is not advisable and suggesting the Mann-Whitney-Wilcoxon or win odds test, so that the condition is not missed in scripted use.

---

rule *Optimal design for a trial with an ordered categorical outcome*

Description

Constructs the decision rule for a one- or two-stage randomised trial with an ordered categorical outcome. This is the recommended entry point to the package: the test statistic and the monitoring scheme are selected by argument, so a single function replaces the eleven separate decision-rule functions provided up to version 1.0.2.

Usage

```
rule(alpha, beta, p1, p2, test = c("S", "M", "W"),
     stopping = c("F", "FS", "none"), criterion = 1, lambda = 1,
     on_degenerate = c("fallback", "warn", "error", "none"), min_n1 = 10)
```

Arguments

alpha	Target type I error rate.
beta	Target type II error rate; power is 1 - beta.
p1	Numeric vector of outcome category probabilities for the control group. Must be non-negative and sum to 1.
p2	Numeric vector of outcome category probabilities for the experimental group. Must be the same length as p1, non-negative, and sum to 1.
test	Test statistic: "S" the score test under the proportional odds model, "M" the Mann-Whitney-Wilcoxon test, "W" the win odds test.
stopping	Monitoring scheme: "F" allows early stopping for futility only, "FS" for futility and superiority, "none" gives a single-stage design.

criterion	Optimality criterion, 1 to 5. 1: minimise the expected total sample size under the null; 2: under the alternative; 3: assuming Pr(H0) = Pr(Ha); 4: balance the two stages prioritising EN0; 5: balance the two stages prioritising the maximum sample size. Ignored when stopping = "none".
lambda	Ratio of experimental to control sample sizes (default 1).
on_degenerate	How to handle a degenerate optimum (see Details). "fallback" (default) warns and returns the single-stage design; "warn" warns and returns the degenerate design; "error" raises an error; "none" returns the raw optimum silently.
min_n1	Smallest stage-1 sample size regarded as usable when detecting degeneracy (default 10).

Details

Minimising the expected sample size under the alternative (criterion = 2) can drive the optimum to a design with a stage-1 sample size of 1 and an interim boundary so far into the lower tail that the interim analysis cannot change any decision: the design is then two-stage in name only. Intuitively, the criterion rewards designs under which continuation to the final analysis is almost certain when the treatment works, so the optimiser commits essentially all information to the final analysis. rule() detects this and, by default, substitutes the corresponding single-stage design, recording the substitution in the returned object rather than making it silently.

Value

An object of class "OptOTrialsRule": a list with components test, stopping, criterion, alpha, beta, lambda, p1, p2, the stage-1 size n1, the interim futility and superiority boundaries t1f and t1s, the maximum size n2, the final boundary t2, a data frame effect giving each boundary on a clinically interpretable effect-size scale (odds ratio for the score test, win odds for the rank-based tests), and the flags degenerate, degenerate_reason and substituted. A print method displays these with labels.

Note

The name rule is also used by the **cli** package, which **MAMS** and several other packages attach. If **cli** is attached after **OptOTrials**, cli::rule masks this function for the rest of the session. Attach **OptOTrials** last, or call OptOTrials::rule() explicitly.

See Also

op, design_table

Examples

```
p1 <- c(0.075, 0.182, 0.319, 0.243, 0.015, 0.166)
p2 <- p2_fun(p1, log(3.06))
rule(0.05, 0.2, p1, p2, test = "M", stopping = "F", criterion = 1)
```

`theta` *Calculation of theta*

Description

This is the function to compute theta (i.e., the expectation of T), which is required for the Win Odds test.

Usage

```
theta(p1, p2)
```

Arguments

`p1`	A vector containing the probabilities of the outcome falling into each level of the control arm.
`p2`	A vector containing the probabilities of the outcome falling into each level of the experimental arm.

Value

The value of theta

Index