

Package ‘caples’

October 8, 2026

Title Fast, Plain-English Readouts for A/B/n Tests

Version 0.1.0

Description Sizes and reads out A/B/n tests on conversion rates in a few function calls. Supports unequal traffic splits, multiple treatment arms with multiplicity correction, sample ratio mismatch checks, achieved minimum detectable effects, and plain-English summaries.

License MIT + file LICENSE

Encoding UTF-8

Imports stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/RobinMahachi/caples>

BugReports <https://github.com/RobinMahachi/caples/issues>

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Robin Mahachi [aut, cre]

Maintainer Robin Mahachi <robinmahachi24@gmail.com>

Repository CRAN

Date/Publication 2026-10-08 17:30:13 UTC

Contents

caples_design	2
caples_test	3
Index	5

caples_design

Size an A/B/n test on a conversion rate

Description

Calculates the sample size needed per arm to detect a given minimum detectable effect (MDE) against a control arm, for one or more treatment arms, with equal or unequal traffic allocation.

Usage

```
caples_design(
  baseline,
  mde,
  mde_type = c("absolute", "relative"),
  power = 0.8,
  arms = 2,
  allocation = NULL,
  conf_level = 0.95,
  alternative = c("two.sided", "greater", "less"),
  adjust_sizing = TRUE
)
```

Arguments

baseline	Expected conversion rate in the control arm (between 0 and 1).
mde	Minimum detectable effect.
mde_type	"absolute" (percentage points, e.g. 0.01 = 5% -> 6%) or "relative" (proportional change, e.g. 0.10 = 5% -> 5.5%).
power	Target power, e.g. 0.8 or 0.85.
arms	Total number of arms including control.
allocation	Optional traffic split, control first, e.g. c(0.5, 0.25, 0.25). Defaults to an equal split. Rescaled to sum to 1.
conf_level	Confidence level; significance level is 1 - conf_level.
alternative	"two.sided", "greater" (treatment > control) or "less" (treatment < control).
adjust_sizing	If TRUE and there is more than one treatment arm, alpha is divided by the number of comparisons (Bonferroni). This is the conservative choice when you plan to use a multiplicity correction such as Holm.

Value

An object of class caples_design: a list with the per-arm sample sizes, total sample size and the inputs used.

Examples

```
caples_design(baseline = 0.05, mde = 0.01)
caples_design(baseline = 0.05, mde = 0.10, mde_type = "relative",
              arms = 3, allocation = c(0.5, 0.25, 0.25))
```

caples_test

*Read out an A/B/n test on a conversion rate***Description**

Compares one or more treatment arms against a control arm. Returns rates, absolute and relative lifts with confidence intervals, raw and multiplicity-adjusted p-values, an optional sample ratio mismatch (SRM) check, a check on whether the test was big enough, and a plain-English summary.

Usage

```
caples_test(
  data,
  group,
  outcome,
  ctrl,
  treatment = NULL,
  trials = NULL,
  conf_level = 0.95,
  adjust = "holm",
  alternative = c("two.sided", "greater", "less"),
  mde = NULL,
  mde_type = c("absolute", "relative"),
  power = 0.8,
  allocation = NULL,
  srm_threshold = 0.001,
  design = NULL
)
```

Arguments

data	A data frame. Either one row per user (with a 0/1 outcome) or one row per arm (with counts, see trials).
group	Name of the column holding the arm labels.
outcome	Name of the outcome column: 0/1 (or TRUE/FALSE) per user, or the number of conversions per arm when trials is supplied.
ctrl	The label of the control arm.
treatment	Label(s) of the treatment arm(s). NULL (default) uses every arm that isn't control.
trials	For aggregated data: name of the column holding the number of users/sends per arm. Leave NULL for row-level data.

conf_level	Confidence level for intervals and significance.
adjust	Multiplicity correction, any method accepted by <code>stats::p.adjust()</code> : "holm", "bonferroni", "BH", "none", ...
alternative	"two.sided", "greater" or "less", where "greater" means treatment > control.
mde	Optional target minimum detectable effect. If supplied, the readout checks whether each arm reached the required sample size.
mde_type	"absolute" or "relative"; see <code>caples_design()</code> .
power	Target power used for the sample-size check and achieved MDE.
allocation	Optional intended traffic split. Either named by arm label or ordered control first, then treatments. Enables the SRM check.
srm_threshold	p-value below which SRM is flagged. The usual convention is 0.001.
design	Optional plan from <code>caples_design()</code> , made before the test. When supplied, mde, mde_type, power, conf_level, alternative and (if not given) allocation are taken from the plan, and the sample-size check compares each arm with the planned n. This is the recommended workflow: decide first, analyse second.

Details

Confidence intervals are two-sided and are not adjusted for multiple comparisons; only p-values are adjusted. The test is a pooled two-proportion z-test, equivalent to `prop.test(correct = FALSE)`.

Value

An object of class `caples_test`.

Examples

```
set.seed(1)
df <- data.frame(
  arm = rep(c("control", "A", "B"), each = 5000),
  converted = c(rbinom(5000, 1, 0.050),
               rbinom(5000, 1, 0.056),
               rbinom(5000, 1, 0.062))
)
# Recommended: plan first, then pass the plan to the readout
plan <- caples_design(baseline = 0.05, mde = 0.01, power = 0.85, arms = 3)
caples_test(df, group = "arm", outcome = "converted", ctrl = "control",
            design = plan)

# Without a plan: supply the targets directly
caples_test(df, group = "arm", outcome = "converted", ctrl = "control",
            mde = 0.01, power = 0.85, allocation = c(1, 1, 1))
```

Index

caples_design, [2](#)
caples_design(), [4](#)
caples_test, [3](#)
stats::p.adjust(), [4](#)