

Package ‘PICOTsize’

October 10, 2026

Type Package

Title Sample Size Calculation for PICOT-Based Study Designs

Version 0.1.0

Description Provides sample size calculators for the study designs covered by the PICOT framework, including cross-sectional, case-control, cohort, superiority, non-inferiority, and equivalence clinical trials, and diagnostic test accuracy studies, following Bhardwaj et al. (2024) [doi:10.4103/jfmpe.jfmpe_1675_23](https://doi.org/10.4103/jfmpe.jfmpe_1675_23). Calculations are performed using the 'epiR' package as a validated computational backend. Includes a 'shiny' application with a PICOT-based design wizard, an interactive sensitivity plot, and automatically generated Methods-section text for manuscripts.

License MIT + file LICENSE

Depends R (>= 4.1.0)

Encoding UTF-8

Imports bslib, epiR, ggplot2, plotly, rlang, shiny

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

URL <https://github.com/AtefehRashidi/PICOTsize>,
<https://orcid.org/0009-0002-3834-3183>

BugReports <https://github.com/AtefehRashidi/PICOTsize/issues>

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Atefeh Rashidi Pour [aut, cre]

Maintainer Atefeh Rashidi Pour <rashidiatefeh98@gmail.com>

Repository CRAN

Date/Publication 2026-10-10 10:10:25 UTC

Contents

apply_dropout	2
calc_casecontrol	3
calc_cohort	4
calc_crosssectional_binary	5
calc_crosssectional_continuous	6
calc_diagnostic_accuracy	7
calc_trial_equivalence	8
calc_trial_noninferiority	9
calc_trial_superiority	10
generate_report_text	11
run_app	12
Index	13

apply_dropout	<i>Adjust a raw sample size for expected dropout</i>
---------------	--

Description

Studies with follow-up, non-response, or attrition need to enrol more participants than the raw formula suggests, so that enough complete observations remain at the end. This function applies the statistically correct inflation formula:

Usage

```
apply_dropout(n, dropout_rate = 0.2)
```

Arguments

- n Numeric. The raw (uninflated) sample size, before accounting for dropout.
- dropout_rate Numeric between 0 and 1 (not inclusive of 1). Expected proportion of participants lost to follow-up or non-response. Defaults to 0.20 (20%), the conventional default used throughout the sample size literature.

Details

$n^* = n / (1 - \text{dropout_rate})$

Note: Bhardwaj et al. (2024), the primary reference for this package, states this same formula in the text, but their worked numeric examples actually use the simpler approximation $n * (1 + \text{dropout_rate})$, which gives a slightly smaller (less conservative) sample size. This package follows the formula as stated in the text, not the arithmetic in the worked examples. See the package Validation vignette for a side-by-side comparison.

Value

A single integer: the dropout-adjusted sample size, rounded up to the nearest whole participant.

Examples

```
apply_dropout(323, dropout_rate = 0.10)
```

calc_casecontrol	<i>Sample size for a case-control study</i>
------------------	---

Description

Estimates the number of cases and controls needed to detect an association between an exposure and an outcome, following Bhardwaj et al. (2024) and using epiR::epi.sccc() as the validated computational backend.

Usage

```
calc_casecontrol(  
  p_exposed_controls,  
  p_exposed_cases,  
  control_case_ratio = 1,  
  power = 0.8,  
  sig_level = 0.05,  
  dropout_rate = 0.2  
)
```

Arguments

p_exposed_controls	Numeric between 0 and 1. Proportion exposed among controls (p0 in the paper’s notation).
p_exposed_cases	Numeric between 0 and 1. Proportion exposed among cases (p1 in the paper’s notation).
control_case_ratio	Numeric >= 1. Number of controls per case. Defaults to 1 (equal numbers of cases and controls).
power	Numeric between 0 and 1. Desired study power. Defaults to 0.80 (80%), matching the paper’s convention.
sig_level	Numeric between 0 and 1. Significance level (alpha). Defaults to 0.05.
dropout_rate	Numeric between 0 and 1. Defaults to 0.20.

Details

Bhardwaj et al. (2024) express the exposure/outcome association in terms of the proportion exposed among cases and among controls (p1 and p0). epiR’s epi.sccc() instead takes an odds ratio (OR). This function converts p1/p0 to an OR internally, so the user can keep thinking in the same terms as the source paper.

Value

A list with n_raw (total, before dropout adjustment), n_final (total, after dropout adjustment), the derived odds ratio, and the inputs used.

Examples

```
# Reproduces the worked lymphoma example in Bhardwaj et al. (2024):
# 25\% exposed in controls, 40\% exposed in cases, equal groups,
# 80\% power, 10\% dropout
calc_casecontrol(p_exposed_controls = 0.25, p_exposed_cases = 0.40,
                 dropout_rate = 0.10)
```

calc_cohort	<i>Sample size for a cohort study</i>
-------------	---------------------------------------

Description

Estimates the number of exposed and unexposed subjects needed to detect a difference in incidence between two groups, following Bhardwaj et al. (2024) and using epiR::epi.sscohortc() as the validated computational backend.

Usage

```
calc_cohort(
  incidence_unexposed,
  incidence_exposed,
  exposed_unexposed_ratio = 1,
  power = 0.8,
  sig_level = 0.05,
  dropout_rate = 0.2
)
```

Arguments

- incidence_unexposed
Numeric between 0 and 1. Expected incidence of the outcome in the unexposed group (p0 in the paper’s notation).
- incidence_exposed
Numeric between 0 and 1. Expected incidence of the outcome in the exposed group (p1 in the paper’s notation).
- exposed_unexposed_ratio
Numeric >= 1. Number of exposed subjects per unexposed subject. Defaults to 1 (equal groups).
- power
Numeric between 0 and 1. Desired study power. Defaults to 0.80 (80%).
- sig_level
Numeric between 0 and 1. Significance level (alpha). Defaults to 0.05.
- dropout_rate
Numeric between 0 and 1. Defaults to 0.20.

Value

A list with n_raw (total, before dropout adjustment), n_final (total, after dropout adjustment), and the inputs used.

Examples

```
# Reproduces the worked air pollution / asthma example in
# Bhardwaj et al. (2024): 20\% incidence unexposed, 30\% incidence
# exposed, equal groups, 80\% power, 10\% dropout
calc_cohort(incidence_unexposed = 0.20, incidence_exposed = 0.30,
            dropout_rate = 0.10)
```

calc_crosssectional_binary
<i>Sample size for a cross-sectional study with a binary outcome</i>

Description

Estimates the sample size needed to estimate a proportion or prevalence in a population, following Bhardwaj et al. (2024) and using epiR::epi.sssimpleestb() as the validated computational backend.

Usage

```
calc_crosssectional_binary(
  p,
  precision,
  error_type = "absolute",
  conf_level = 0.95,
  dropout_rate = 0.2
)
```

Arguments

p	Numeric between 0 and 1. Expected proportion/prevalence in the population (from prior studies or a pilot study).
precision	Numeric. Acceptable margin of error (absolute, in the same 0-1 scale as p, unless error_type = "relative").
error_type	Character. Either "absolute" or "relative". Defaults to "absolute", matching the worked examples in Bhardwaj et al. (2024).
conf_level	Numeric. Confidence level, e.g. 0.95 for 95%.
dropout_rate	Numeric between 0 and 1. Expected dropout / non-response rate. Defaults to 0.20. Set to 0 to skip dropout adjustment entirely.

Value

A list with the raw sample size (n_raw), the dropout-adjusted sample size (n_final), and the inputs used, so the result can be fed directly into report generation.

Examples

```
# Reproduces the worked example in Bhardwaj et al. (2024):
# prevalence 30%, 5% absolute precision, 95% CI, 10% dropout
calc_crosssectional_binary(p = 0.30, precision = 0.05,
                           dropout_rate = 0.10)
```

calc_crosssectional_continuous
<i>Sample size for a cross-sectional study with a continuous outcome</i>

Description

Estimates the sample size needed to estimate a population mean, following Bhardwaj et al. (2024) and using epiR::epi.sssimpleestc() as the validated computational backend.

Usage

```
calc_crosssectional_continuous(
  mean,
  sd,
  precision,
  error_type = "absolute",
  conf_level = 0.95,
  dropout_rate = 0.2
)
```

Arguments

mean	Numeric. Expected mean of the outcome (from prior studies or a pilot study). Required by the epiR backend even when error_type = "absolute".
sd	Numeric. Expected standard deviation of the outcome.
precision	Numeric. Acceptable margin of error. In the same units as sd if error_type = "absolute", or as a fraction of mean if error_type = "relative".
error_type	Character. Either "absolute" or "relative". Defaults to "absolute", matching the worked examples in Bhardwaj et al. (2024). Note this differs from epiR's own default ("relative") – we override it here.
conf_level	Numeric. Confidence level, e.g. 0.95 for 95%.
dropout_rate	Numeric between 0 and 1. Defaults to 0.20.

Value

A list with the same structure as [calc_crosssectional_binary()].

Examples

```
# Reproduces the worked SBP example in Bhardwaj et al. (2024):
# mean SBP unspecified in the paper's formula, SD = 3 mmHg,
# precision = 0.5 mmHg, 95% CI, 10% dropout
calc_crosssectional_continuous(mean = 120, sd = 3, precision = 0.5,
                               dropout_rate = 0.10)
```

calc_diagnostic_accuracy	<i>Sample size to estimate the sensitivity and specificity of a diagnostic test</i>
--------------------------	---

Description

Estimates the sample size needed to evaluate a diagnostic test’s accuracy, using epiR::epi.ssdxsesp() as the validated computational backend. That function implements the method of Buderer (1996) and Hajian-Tilaki (2014), which calculates the required n for sensitivity and for specificity separately, then takes the LARGER of the two as the total study sample size – because the same group of subjects is used to estimate both simultaneously, not two separate samples.

Usage

```
calc_diagnostic_accuracy(
  expected_sensitivity,
  expected_specificity,
  prevalence,
  precision,
  error_type = "absolute",
  conf_level = 0.95,
  dropout_rate = 0
)
```

Arguments

expected_sensitivity	Numeric between 0 and 1. Prior estimate of the test’s sensitivity.
expected_specificity	Numeric between 0 and 1. Prior estimate of the test’s specificity.
prevalence	Numeric between 0 and 1. Expected prevalence of the disease/condition in the study population.
precision	Numeric. Acceptable margin of error for the estimate (absolute, unless error_type = "relative").

error_type	Character. Either "absolute" or "relative". Defaults to "absolute", matching the worked example in Bhardwaj et al. (2024).
conf_level	Numeric. Confidence level, e.g. 0.95 for 95%.
dropout_rate	Numeric between 0 and 1. Defaults to 0, since Bhardwaj et al. (2024) do not apply a dropout adjustment to their diagnostic test example. Set to a positive value if your study design calls for one.

Details

Note this differs from Bhardwaj et al. (2024), who calculate the two requirements separately and add them together. Summing effectively assumes two independent studies, which is not how diagnostic accuracy studies are actually run. This package follows the epiR/Buderer approach (the larger of the two), as the methodologically correct one. See the package Validation vignette for a side-by-side comparison.

Value

A list with `n_sensitivity`, `n_specificity`, `n_raw` (the larger of the two – the actual required enrolment), `n_final` (dropout-adjusted), and the inputs used.

Examples

```
# Reproduces the worked hypertension test example in
# Bhardwaj et al. (2024): sensitivity 80%, specificity 90%,
# prevalence 20%, 5% absolute margin of error
calc_diagnostic_accuracy(expected_sensitivity = 0.80,
                          expected_specificity = 0.90,
                          prevalence = 0.20, precision = 0.05)
```

calc_trial_equivalence

Sample size for an equivalence trial (binary outcome)

Description

Estimates the sample size needed to demonstrate that two treatments are, for practical purposes, equally effective, using `epiR::epi.ssequb()` as the validated computational backend.

Usage

```
calc_trial_equivalence(
  p_standard,
  p_new,
  delta,
  power = 0.8,
  sig_level = 0.05,
```

```

    treat_control_ratio = 1,
    dropout_rate = 0.2
  )

```

Arguments

p_standard	Numeric between 0 and 1. Outcome rate in the standard/control treatment group.
p_new	Numeric between 0 and 1. Outcome rate in the new treatment group.
delta	Numeric ≥ 0 . Equivalence limit – the maximum difference in either direction still considered "equivalent".
power	Numeric between 0 and 1. Desired study power. Defaults to 0.80.
sig_level	Numeric between 0 and 1. Significance level (alpha). Defaults to 0.05.
treat_control_ratio	Numeric ≥ 1 . Number in the treatment group per subject in the control group. Defaults to 1.
dropout_rate	Numeric between 0 and 1. Defaults to 0.20.

Value

A list with n_raw, n_final, and the inputs used.

Examples

```

calc_trial_equivalence(p_standard = 0.45, p_new = 0.45,
                      delta = 0.10)

```

calc_trial_noninferiority

Sample size for a non-inferiority trial (binary outcome)

Description

Estimates the sample size needed to demonstrate that a new treatment is not unacceptably worse than a standard treatment, using `epiR::epi.ssnnfb()` as the validated computational backend.

Usage

```

calc_trial_noninferiority(
  p_standard,
  p_new,
  delta,
  power = 0.8,
  sig_level = 0.05,
  treat_control_ratio = 1,
  dropout_rate = 0.2
)

```

Arguments

p_standard	Numeric between 0 and 1. Outcome rate in the standard/control treatment group.
p_new	Numeric between 0 and 1. Outcome rate in the new treatment group.
delta	Numeric ≥ 0 . Non-inferiority margin – the maximum acceptable drop in outcome rate for the new treatment to still be considered non-inferior. This is the parameter most often mis-specified, so double-check it reflects a clinically (not just statistically) meaningful difference.
power	Numeric between 0 and 1. Desired study power. Defaults to 0.80.
sig_level	Numeric between 0 and 1. Significance level (alpha). Defaults to 0.05.
treat_control_ratio	Numeric ≥ 1 . Number in the treatment group per subject in the control group. Defaults to 1.
dropout_rate	Numeric between 0 and 1. Defaults to 0.20.

Value

A list with n_raw, n_final, and the inputs used.

Examples

```
calc_trial_noninferiority(p_standard = 0.45, p_new = 0.45,
                          delta = 0.10)
```

```
calc_trial_superiority
```

Sample size for a superiority trial (binary outcome)

Description

Estimates the sample size needed to demonstrate that a new treatment is better than a standard treatment, following Bhardwaj et al. (2024) and using epiR::epi.sssupb() as the validated computational backend.

Usage

```
calc_trial_superiority(
  p_standard,
  p_new,
  delta,
  power = 0.8,
  sig_level = 0.05,
  sided_test = 2,
  treat_control_ratio = 1,
  dropout_rate = 0.2
)
```

Arguments

p_standard	Numeric between 0 and 1. Outcome rate in the standard/control treatment group.
p_new	Numeric between 0 and 1. Outcome rate in the new treatment group.
delta	Numeric ≥ 0 . Superiority margin – the minimum difference researchers want to be able to detect.
power	Numeric between 0 and 1. Desired study power. Defaults to 0.80.
sig_level	Numeric between 0 and 1. Significance level (alpha). Defaults to 0.05.
sided_test	Either 1 or 2. Bhardwaj et al. (2024) use a one-sided test for superiority trials (the traditional convention). However, epiR's own documentation notes that "regulatory agencies and most clinical trial guidelines recommend two-sided tests for superiority trials" – this is a genuine, unresolved difference of opinion in the literature, not a bug. Defaults to 2 (two-sided), matching current regulatory guidance; set to 1 to reproduce the paper's own worked example.
treat_control_ratio	Numeric ≥ 1 . Number in the treatment group per subject in the control group. Defaults to 1.
dropout_rate	Numeric between 0 and 1. Defaults to 0.20.

Value

A list with n_raw, n_final, and the inputs used.

Examples

```
# Reproduces the worked cancer survival example in
# Bhardwaj et al. (2024) -- use sided_test = 1 to match their
# one-sided convention exactly
calc_trial_superiority(p_standard = 0.45, p_new = 0.61,
                      delta = 0.10, sided_test = 1)
```

generate_report_text	<i>Generate a Methods-section paragraph from a sample size result</i>
----------------------	---

Description

Takes the list returned by any of the calc_* functions in this package and produces a ready-to-use paragraph describing the sample size calculation, written in the style of a Methods section, following Bhardwaj et al. (2024)'s own reporting conventions.

Usage

```
generate_report_text(result)
```

Arguments

result A list, as returned by `calc_crosssectional_binary()`, `calc_crosssectional_continuous()`, `calc_casecontrol()`, `calc_cohort()`, `calc_trial_superiority()`, `calc_trial_noninferiority()`, `calc_trial_equivalence()`, or `calc_diagnostic_accuracy()`.

Value

A single character string containing the report paragraph.

Examples

```
result <- calc_crosssectional_binary(p = 0.30, precision = 0.05,  
                                   dropout_rate = 0.10)  
generate_report_text(result)
```

run_app

Launch the PICOTsize Shiny app

Description

Launch the PICOTsize Shiny app

Usage

```
run_app()
```

Value

No return value. Launches the Shiny app in a browser or the RStudio Viewer pane.

Examples

```
if (interactive()){  
  run_app()  
}
```

Index

`apply_dropout`, [2](#)

`calc_casecontrol`, [3](#)

`calc_cohort`, [4](#)

`calc_crosssectional_binary`, [5](#)

`calc_crosssectional_continuous`, [6](#)

`calc_diagnostic_accuracy`, [7](#)

`calc_trial_equivalence`, [8](#)

`calc_trial_noninferiority`, [9](#)

`calc_trial_superiority`, [10](#)

`generate_report_text`, [11](#)

`run_app`, [12](#)