Package ‘anovapowersim’

May 31, 2026

Title Simple Power Simulations for ANOVAs

Version 1.1.0

Description A-priori power simulations and power-calculations for within, between and mixed ANOVAs based on target (partial) eta-squared values. Supports complex designs with more than two factors and their interactions with a single function call.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports car, dplyr, future, future.apply, ggplot2, MASS, purrr, rlang, stats, tibble, tidyr

Suggests knitr, pkgdown, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://shaheedazaad.github.io/anovapowersim/, https://github.com/shaheedazaad/anovapowersim

BugReports https://github.com/shaheedazaad/anovapowersim/issues

NeedsCompilation no

Author Shaheed Azaad [aut, cre]

Maintainer Shaheed Azaad <sazaad@uni-muenster.de>

Repository CRAN

Date/Publication 2026-05-31 15:00:38 UTC

Contents

`balanced_anova_design` *Create a balanced factorial ANOVA design specification*

Description

Builds the design object used by `power_curve()`, `design_term_means()`, and `simulate_design_dataset()`. This object stores factor names, level counts, generated factor levels, and the between/within cell grids.

Usage

```
balanced_anova_design(between = NULL, within = NULL)
```

Arguments

`between`	Named integer vector of between-subject factor level counts, e.g. `c(group = 2)`. Use `NULL` for no between-subject factors.
`within`	Named integer vector of within-subject factor level counts, e.g. `c(time = 3)`. Use `NULL` for no within-subject factors.

Value

An object of class `anovapowersim_design_spec`.

Examples

```
d <- balanced_anova_design(between = c(group = 2), within = c(time = 3))
d$between_cells
d$within_cells
```

compute_scale_factor *Compute the mean-deviation scaling factor from a change in partial eta squared*

Description

Given an existing partial eta squared for a term and a target partial eta squared, returns the multiplier k that must be applied to that term's additive contribution to the cell means in order to obtain the target effect size under the same residual structure.

Usage

```
compute_scale_factor(old_pes, new_pes)
```

Arguments

old_pes	Numeric scalar in (0, 1), or a numeric-looking character scalar such as ".310". The current partial eta squared for the term of interest.
new_pes	Numeric scalar in (0, 1), or a numeric-looking character scalar such as ".200". The target partial eta squared.

Details

The derivation is straightforward: partial eta squared can be written as `pes = SS_effect / (SS_effect + C)`, where `C` is the part of the denominator held fixed by this package's rescaling. Thus `pes / (1 - pes)` scales as the target effect's sum of squares. Scaling the term's deviations by `k` scales the target effect's sum of squares by `k^2`, so the required multiplier is

$$k = \sqrt{\frac{p_{\text{new}}/(1 - p_{\text{new}})}{p_{\text{old}}/(1 - p_{\text{old}})}}.$$

Value

A single positive numeric value `k`. `k > 1` amplifies the effect, `k < 1` shrinks it, and `k == 1` leaves it unchanged.

See Also

`design_term_means()`, `power_curve()`

Examples

```
compute_scale_factor(0.10, 0.05)   # shrink
compute_scale_factor(0.05, 0.10)   # amplify
```

design_term_means — *Build calibrated default means for a design term*

Description

Creates the default contrast pattern for one ANOVA term and scales it so an exact reference dataset has the requested partial eta squared under the supplied balanced design assumptions.

Usage

```
design_term_means(
  design,
  term,
  target_pes,
  n,
  sd = 1,
  r = 0.5,
  gpower = FALSE,
  ss_type = "III"
)
```

Arguments

design	An anovapowersim_design_spec from balanced_anova_design().
term	Character scalar naming the ANOVA term to target. Interaction terms are order-insensitive.
target_pes	Target partial eta squared.
n	Sample size per between-subject cell. For pure within designs, this is the total sample size.
sd	Common outcome standard deviation.
r	Compound-symmetric correlation among within-subject cells.
gpower	Logical; if TRUE, calibrate to the G*Power-style noncentrality convention lambda = total_n * f^2 (using the 'as in Cohen (1988) option for within-subjects designs).
ss_type	Sums-of-squares type for the tested ANOVA term. "III" is the default for order-invariant tests in unbalanced designs. Use "I" to reproduce sequential stats::aov() tests.

Value

A numeric matrix of cell means, with rows indexing between cells and columns indexing within cells.

Examples

```
d <- balanced_anova_design(between = c(group = 2), within = c(time = 2))
design_term_means(d, term = "group:time", target_pes = 0.2, n = 20)
```

---

plot_power_curve *Plot a simulation-based power curve*

Description

Renders an `anovapowersim_curve` as a `ggplot2` line + ribbon with a horizontal reference at requested power values and, when auto-search was used, a vertical marker at the estimated required total sample size.

Usage

```
plot_power_curve(
  x,
  show_target = TRUE,
  power_lines = NULL,
  show_n_needed = TRUE,
  ...
)
```

Arguments

`x`	An `anovapowersim_curve` object from `power_curve()`.
`show_target`	Logical; draw the horizontal target power line (default `TRUE`).
`power_lines`	Optional numeric vector of additional power reference lines, e.g. `c(.80, .90)`.
`show_n_needed`	Logical; draw the vertical line at `n_needed` (default `TRUE`).
`...`	Unused, for S3 consistency.

Value

A `ggplot` object.

See Also

`power_curve()`

Examples

```
pc <- power_curve(
  between = c(group = 2),
  within = c(time = 2),
  term = "group:time",
  target_pes = 0.2,
  n_range = c(20, 30),
  n_sims = 1000,
  seed = 123
)
plot_power_curve(pc)
```

`power_curve` *Simulate ANOVA power from a balanced factorial design*

Description

Simulation-based power estimation for balanced factorial designs under sphericity. Users specify the between- and within-subject factors, the ANOVA term to test, a target partial eta squared, and explicit sample sizes. The function creates a default contrast pattern for the target term, scales it to the requested partial eta squared, simulates datasets, refits `stats::aov()`, and estimates power by counting `p` < `alpha`.

Usage

```
power_curve(
  between = NULL,
  within = NULL,
  term,
  target_pes,
  n_range,
  n_sims = 10000,
  alpha = 0.05,
  ss_type = "III",
  gpower = FALSE,
  progress = interactive(),
  parallel = FALSE,
  cores = NULL,
  seed = NULL
)
```

Arguments

`between` Named integer vector of between-subject factor level counts, e.g. `c(group = 2)`. Use `NULL` for no between-subject factors.

within	Named integer vector of within-subject factor level counts, e.g. c(time = 3, condition = 4). Use NULL for no within-subject factors.
term	Character scalar naming the ANOVA term to test, e.g. "group:time". Interaction terms are order-insensitive; "time:group" resolves to "group:time" when that is the design's factor order.
target_pes	Target partial eta squared for term.
n_range	Integer vector of sample sizes per between-subject cell. For pure within-subject designs, this is the total sample size.
n_sims	Number of simulated datasets per sample size.
alpha	Significance threshold.
ss_type	Sums-of-squares type for the tested ANOVA term. "III" is the default for order-invariant tests in unbalanced designs. Use "I" to reproduce sequential stats::aov() tests.
gpower	Logical; if TRUE, calibrate means to the G*Power-style noncentrality convention lambda = total_n * f^2. The default FALSE calibrates the empirical reference dataset to target_pes, equivalent to lambda = den_df * f^2 for the fitted ANOVA.
progress	Logical; if TRUE, show a text progress bar.
parallel	Logical; if TRUE, run simulations for each sample size via the future ecosystem.
cores	Optional positive integer number of cores to use when parallel = TRUE. If NULL, uses one fewer than the number of available cores, with a minimum of one.
seed	Optional integer seed for reproducibility.

Value

An anovapowersim_curve object. The $results tibble contains n_per_cell, total_n, n_sims, numerator and denominator degrees of freedom (num_df, den_df), the noncentrality parameter (ncp), calculated power (power_calc), and simulated power (power_sim).

Examples

```
power_curve(
  between = c(cond = 2),
  within = c(stim = 2),
  term = "cond:stim",
  target_pes = 0.14,
  n_range = c(16, 20, 23, 28), # n per between-subject cell
  n_sims = 1000,
  seed = 123
)

power_curve(
  between = c(group = 2),
  within = c(time = 2),
  term = "group:time",
```

```
  target_pes = 0.14,
  n_range = c(12, 16, 20),
  n_sims = 5000,
  parallel = TRUE,
  cores = 4,
  seed = 123
)
```

---

power_n *Search for the sample size needed for target ANOVA power*

Description

Adaptive simulation search for the per-between-cell sample size needed to reach a requested power for a balanced factorial ANOVA design. The search doubles upward from `n_start` until it brackets the target or reaches `n_max`, then refines the bracket using interpolation with midpoint bisection as a fallback.

Usage

```
power_n(
  between = NULL,
  within = NULL,
  term,
  target_pes,
  power = 0.9,
  n_sims = 10000,
  alpha = 0.05,
  ss_type = "III",
  n_start = NULL,
  n_max = 1000,
  tol = 0.03,
  gpower = FALSE,
  progress = interactive(),
  parallel = FALSE,
  cores = NULL,
  seed = NULL
)
```

Arguments

`between`	Named integer vector of between-subject factor level counts, e.g. `c(group = 2)`. Use `NULL` for no between-subject factors.
`within`	Named integer vector of within-subject factor level counts, e.g. `c(time = 3, condition = 4)`. Use `NULL` for no within-subject factors.

`term`	Character scalar naming the ANOVA term to test, e.g. `"group:time"`. Interaction terms are order-insensitive; `"time:group"` resolves to `"group:time"` when that is the design's factor order.
`target_pes`	Target partial eta squared for `term`.
`power`	Desired target power.
`n_sims`	Number of simulated datasets per sample size.
`alpha`	Significance threshold.
`ss_type`	Sums-of-squares type for the tested ANOVA term. `"III"` is the default for order-invariant tests in unbalanced designs. Use `"I"` to reproduce sequential `stats::aov()` tests.
`n_start`	Starting sample size per between-subject cell. If `NULL`, starts at the smallest value that can support empirical calibration for the requested design.
`n_max`	Maximum sample size per between-subject cell.
`tol`	Acceptable precision above target power. If no simulated value at or above `power` is also no more than `power + tol`, `power_n()` warns that the requested precision band was not reached.
`gpower`	Logical; if `TRUE`, calibrate means to the G*Power-style noncentrality convention `lambda = total_n * f^2`. The default `FALSE` calibrates the empirical reference dataset to `target_pes`, equivalent to `lambda = den_df * f^2` for the fitted ANOVA.
`progress`	Logical; if `TRUE`, show a text progress bar.
`parallel`	Logical; if `TRUE`, run simulations for each sample size via the `future` ecosystem.
`cores`	Optional positive integer number of cores to use when `parallel = TRUE`. If `NULL`, uses one fewer than the number of available cores, with a minimum of one.
`seed`	Optional integer seed for reproducibility.

Value

An `anovapowersim_curve` object with `n_needed` and `total_n_needed`. For `power_n()`, `n_needed` is always an explicitly simulated `n_per_cell` value, never an interpolated sample size. If the search reaches target power but no simulated value lands inside `[power, power + tol]`, `power_n()` reports the smallest explicitly simulated value at or above target power and warns that the requested precision band was not reached.

Examples

```
power_n(
  between = c(cond = 2),
  within = c(stim = 4),
  term = "cond:stim",
  target_pes = 0.14,
  alpha = 0.05,
  power = 0.90,
  n_sims = 1000, # use 5000+ for a more precise estimate
```

```
  seed = 123 # for reproducibility
)
```

print.anovapowersim_curve
Print an anovapowersim power curve

Description

Compact one-screen summary: target, term, effective effect size, estimated per-cell and total sample sizes, and the first and last rows of the power curve.

Usage

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

Arguments

x	An anovapowersim_curve object.
...	Unused.

Value

Invisibly returns x.

simulate_design_dataset
Simulate data from a balanced ANOVA design

Description

Generates one long-format dataset from a balanced design. Supply means from design_term_means() or any conformable matrix with one row per between-subject cell and one column per within-subject cell.

Usage

```
simulate_design_dataset(design, n, means, sd = 1, r = 0.5, empirical = FALSE)
```

Arguments

design	An anovapowersim_design_spec from balanced_anova_design().
n	Sample size per between-subject cell. For pure within designs, this is the total sample size.
means	Numeric matrix of population cell means.
sd	Common outcome standard deviation.
r	Compound-symmetric correlation among within-subject cells.
empirical	Logical; if TRUE, use MASS::mvrnorm(empirical = TRUE) so the generated sample closely matches the requested means/covariance.

Value

A tibble ready for stats::aov() with columns id, factor columns, and value.

Examples

```
d <- balanced_anova_design(between = c(group = 2), within = c(time = 2))
m <- design_term_means(d, term = "group:time", target_pes = 0.2, n = 20)
sim <- simulate_design_dataset(d, n = 20, means = m)
head(sim)
```

---

summary.anovapowersim_curve

Summarise an anovapowersim power curve

Description

Returns the full $results tibble along with a small header containing the target, effective effect size, and estimated n_needed.

Usage

```
## S3 method for class 'anovapowersim_curve'
summary(object, ...)
```

Arguments

object	An anovapowersim_curve object.
...	Unused.

Value

A list with elements header (named character) and curve (tibble), invisibly; printed to console as well.

Index