Package: anovapowersim (via r-universe)

October 9, 2026

Title Simple Power Simulations for ANOVAs

Version 1.2.0

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

Roxygen list(markdown = TRUE)

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, https://shaheedazaad.r-universe.dev/anovapowersim

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

Config/pak/sysreqs cmake make libicu-dev

Repository https://shaheedazaad.r-universe.dev

Date/Publication 2026-09-09 04:29:57 UTC

RemoteUrl https://github.com/shaheedazaad/anovapowersim

RemoteRef v1.2.0

RemoteSha c5cfee2f2de5a3a3dcf384c28b6e32da44e5db46

Contents

balanced_anova_design . . . 2
cell_design . . . 3
compute_scale_factor . . . 4
design_term_means . . . 5
f_to_pes . . . 7
means_pattern . . . 8
plot_power_curve . . . 9
power_achieved . . . 10
power_achieved_calc . . . 12
power_curve . . . 14
power_n . . . 17
power_n_calc . . . 20
power_sensitivity . . . 21
power_sensitivity_calc . . . 24
power_unbalanced . . . 26
print.anovapowersim_achieved_power . . . 28
print.anovapowersim_curve . . . 28
print.anovapowersim_sensitivity . . . 29
print.anovapowersim_unbalanced_power . . . 30
simulate_design_dataset . . . 30
summary.anovapowersim_achieved_power . . . 31
summary.anovapowersim_curve . . . 32
summary.anovapowersim_sensitivity . . . 32
summary.anovapowersim_unbalanced_power . . . 33
unbalanced_covariance . . . 33
within_covariance . . . 34

Index **36**

`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
```

`cell_design` *Define cells for a means-based unbalanced ANOVA design*

Description

Creates the complete cell table used by `power_unbalanced()`. Each cell is defined by its factor levels, sample size (`n`), and population mean (`m`). End each cell after both reserved values have been supplied. The common population standard deviation belongs in `unbalanced_covariance()`.

Usage

```
cell_design(..., within = NULL, default_n = NULL, default_m = NULL)
```

Arguments

`...`	Repeated named cell definitions. Each cell must contain the same factor names in the same order, plus `n` and `m`. Every factor must have at least 2 observed levels, and every combination of factor levels must appear exactly once (or be filled automatically; see `default_n`).
`within`	Character vector naming factors in `...` that are measured within subjects, or `NULL` for a purely between-subject design. Stored on the returned object and read by `power_unbalanced()`. Within-cell names used by `unbalanced_covariance()` join level values with `_`; these names must be unique, and within-factor levels must not contain `:`.
`default_n`, `default_m`	Optional scalars used to fill any missing cells in the complete factorial design. Supply both to auto-fill missing cells with these values; supply none to require every cell to be defined explicitly (the default). Supplying only one is an error. When cells are auto-filled, a message reports their count and exact factor-level combinations so that unintended levels can be spotted.

Details

The `m` values are literal population cell means: their magnitudes and all effects they contain are used as supplied. This differs from `means_pattern()`, whose values specify only a relative shape that balanced simulation functions project onto the tested term, normalize, and rescale to `target_pes`.

Value

An `anovapowersim_cell_design` tibble with one row per design cell.

Lifecycle

Experimental

`cell_design()` is experimental and is available only in the development version of `anovapowersim`. Its API may change.

See Also

`means_pattern()` for shape-only patterns used by balanced simulation functions.

Examples

```
design <- cell_design(
  group = "control", time = "pre",  n = 22, m = 10.0,
  group = "control", time = "post", n = 22, m = 11.0,
  group = "treatment", time = "pre",  n = 31, m = 10.1,
  group = "treatment", time = "post", n = 31, m = 12.4,
  within = "time"
)
design
```

`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 means for a design term*

Description

Projects a supplied relative pattern (or uses the documented linear/Kronecker default) 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",
  means_pattern = NULL
)
```

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 *(as in the "Cohen (1988)" option for within-subjects designs in G*Power). G*Power's estimates can differ from target_pes, especially for small samples or terms with more degrees of freedom; a warning is issued when gpower = TRUE. The default gpower = FALSE is recommended.
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.
means_pattern	Optional relative mean shape from means_pattern(). Sparse values are projected onto term before calibration. If NULL, the deterministic linear/Kronecker default is used.

Value

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

Covariance limitation

This manual helper does not accept within_covariance() specifications. Calibration always uses the compound-symmetric covariance defined by sd and r. Consequently, its calibrated means can differ from those used by power_curve(), power_n(), or power_achieved() with a custom covariance, because the covariance affects the reference residual sum of squares and therefore the mean scale factor.

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)
```

f_to_pes *Convert Cohen's f to partial eta squared*

Description

Converts Cohen's f for an effect to its corresponding partial eta squared:

$$\eta_p^2 = \frac{f^2}{1 + f^2}.$$

The same formula applies to between-subject, within-subject, and interaction effects when f uses the error variance corresponding to that effect's partial eta squared. This helper does not translate between G*Power's repeated-measures effect-size conventions.

Usage

```
f_to_pes(f)
```

Arguments

f	A single finite, nonnegative numeric value representing Cohen's f, not an ANOVA F statistic or f-squared.

Value

A single numeric partial eta squared. Zero maps to zero; power functions retain their own restrictions on `target_pes`. Very large f values may yield exactly one because of floating-point rounding.

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Lawrence Erlbaum Associates.

See Also

`compute_scale_factor()`, `power_n_calc()`

Examples

```
f_to_pes(0.25) # approximately 0.05882
power_n_calc(
  between = c(group = 2),
  term = "group",
  target_pes = f_to_pes(0.25)
)
```

`means_pattern` *Define a sparse relative cell-mean pattern*

Description

Creates a sparse mean-shape specification for the balanced simulation functions. End each cell definition with `value`. Unlisted cells have raw value zero, and factors in the tested term that are omitted from every row are broadcast when the pattern is resolved against a design.

Usage

```
means_pattern(...)
```

Arguments

`...`	Repeated named sparse-cell definitions. Each definition must use the same factor names in the same order and end in a finite numeric scalar named `value`. Factor levels may be supplied as one-based integer indices (for example, `time = 3`) or as the generated balanced-design names (for example, `time = "time3"`). The two forms are equivalent after the pattern is resolved against a design.

Details

Pattern values describe relative shape, not effect magnitude. The selected power function projects the raw values onto the requested ANOVA term, normalizes that component, and rescales it uniformly to reach `target_pes`. Multiplying all values by one positive constant, adding an intercept or a lower-order component, or reversing every sign therefore leaves the same target-term direction (up to sign). Under nonsphericity, different directions within a multi-df term can nevertheless produce different simulated power.

This differs from `cell_design()`, where each `m` is a literal population mean whose magnitude directly determines the simulated effect.

Value

An object of class `anovapowersim_means_pattern`, retaining the sparse definitions until a balanced simulation function resolves them against its design and tested term.

Default direction

When no pattern is supplied, balanced simulations use centered scores in generated level order, `i - (L + 1) / 2` for levels `i = 1, ..., L`, normalized to unit length. Interactions use the Kronecker product of their factors' normalized score vectors, followed by one final normalization after broadcasting. This is an ordered, reproducible convention rather than a neutral scientific assumption; an explicit pattern is recommended whenever the expected shape is known.

See Also

`cell_design()` for unbalanced designs with literal cell means.

Examples

```
trend <- means_pattern(
  time = 1, value = 0,
  time = 2, value = 0.3,
  time = 3, value = 0.5,
  time = 4, value = 0.6
)

interaction_shape <- means_pattern(
  group = "group1", time = "time3", value = 1,
  group = "group2", time = "time3", value = -1
)
```

---

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_achieved *Estimate achieved ANOVA power at a fixed sample size*

Description

Simulates one balanced factorial ANOVA design point for a fixed partial eta squared and sample size. The simulation estimate is the primary achieved power result; noncentral-F calculated power is retained as a diagnostic.

Usage

```
power_achieved(
  between = NULL,
  within = NULL,
  term,
  target_pes,
  n,
  n_sims = 10000,
  alpha = 0.05,
  ss_type = "III",
  gpower = FALSE,
  progress = interactive(),
  parallel = FALSE,
  cores = NULL,
  seed = NULL,
  covariance = NULL,
  means_pattern = NULL,
  sim_correction = c("auto", "GG", "none")
)
```

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 — Number of subjects per between-subject cell. For a purely within-subject design, 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. Greenhouse–Geisser-corrected simulated p-values are available only for `"III"` and `"II"`.

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. G*Power's estimates can differ from `target_pes`, especially for small samples or terms with more degrees of freedom; a warning is issued when `gpower = TRUE`. The default `gpower = FALSE` is recommended.

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.

covariance — Optional within-subject covariance specification created by `within_covariance()`. Raw covariance matrices are not accepted, which avoids silently assuming a within-cell order. The default `NULL` uses standard deviations of `1` and a compound-symmetric correlation of `0.5` and issues a warning stating those defaults. A `within_covariance()` specification issues a warning when correlation pairs are omitted: its `default_correlation` applies only to those undefined pairs, while explicitly defined correlations are unchanged. All measurements use the specification's common marginal variance, while unequal correlations remain supported. For terms containing within-subject factors, the resolved covariance matrix is also used to derive a term-specific population Greenhouse–Geisser epsilon for `power_calc`. With the default `sim_correction = "auto"`, a population epsilon below `1 - 1e-8` also selects Greenhouse–Geisser-corrected simulated p-values for `ss_type` `"II"` or `"III"`.

means_pattern — Optional relative cell-mean shape created by `means_pattern()`. The sparse values are projected onto `term`, normalized, and uniformly rescaled to reach `target_pes`. If `NULL`, simulations use the package's deterministic linear/Kronecker pattern. For multi-df nonspherical within-subject terms, simulated power is conditional on this direction, so an explicit pattern is recommended when the expected shape is known.

`sim_correction` Sphericity correction for simulated p-values: `"auto"` (the default) uses Greenhouse–Geisser correction when the term-specific population epsilon is below `1 - 1e-8` and `ss_type` is `"II"` or `"III"`; `"GG"` requests correction for every simulated dataset; and `"none"` always uses the uncorrected univariate test. `"GG"` is an error with `ss_type = "I"`. For a between-only term or a within component with one degree of freedom, `"GG"` silently resolves to `"none"` because no sphericity correction applies.

Value

An `anovapowersim_achieved_power` object. `$results` contains the standard one-row power diagnostics. `$achieved_power` is the simulated power estimate and `$calculated_power` is the calculated-power diagnostic.

Lifecycle

Experimental

`power_achieved()` is experimental and is available only in the development version of `anovapowersim`. Its API and reporting format may change.

Examples

```
power_achieved(
  between = c(group = 2),
  within = c(time = 2),
  term = "group:time",
  target_pes = 0.14,
  n = 20,
  n_sims = 100,
  seed = 123
)
```

`power_achieved_calc` *Calculate achieved ANOVA power at a fixed sample size*

Description

Calculation-only counterpart to `power_achieved()`. Numerator and denominator degrees of freedom, noncentrality, and achieved power are calculated directly without simulating data or fitting ANOVA models.

Usage

```
power_achieved_calc(
  between = NULL,
  within = NULL,
  term,
  target_pes,
  n,
  alpha = 0.05,
  gpower = FALSE,
  epsilon = 1
)
```

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`	Number of subjects per between-subject cell. For a purely within-subject design, this is the total sample size.
`alpha`	Significance threshold.
`gpower`	Logical; if `TRUE`, use the G*Power-style noncentrality convention* `lambda = total_n * f^2`. *The default* `FALSE` *uses* `lambda = den_df * f^2`. *G*Power's estimates can differ from `target_pes`, especially for small samples or terms with more degrees of freedom; a warning is issued when `gpower = TRUE`. The default `gpower = FALSE` is recommended.
`epsilon`	Population nonsphericity correction for the within-subject component of `term`. Must lie between the theoretical lower bound `1 / within_term_df` and `1`. The default `1` assumes sphericity. Values below `1` multiply the numerator degrees of freedom, denominator degrees of freedom, and noncentrality parameter. Nonsphericity corrections do not apply to purely between-subject terms.

Value

An `anovapowersim_achieved_power` object. `$achieved_power` and `$calculated_power` contain the calculated-power estimate. In `$results`, simulation-specific result columns are `NA` because no simulations are run.

Lifecycle

Experimental

`power_achieved_calc()` is experimental and is available only in the development version of `anovapowersim`. Its API and reporting format may change.

Examples

```
power_achieved_calc(
  between = c(group = 2),
  within = c(time = 3),
  term = "group:time",
  target_pes = 0.08,
  n = 30,
  gpower = TRUE,
  epsilon = 0.80
)
```

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

Description

Simulation-based power estimation for balanced factorial designs. Users specify the between- and within-subject factors, the ANOVA term to test, a target partial eta squared, and explicit sample sizes. The function projects an explicit relative means pattern (or uses the documented linear/Kronecker default), scales it to the requested partial eta squared, simulates datasets, refits the ANOVA, 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,
  covariance = NULL,
  means_pattern = NULL,
  sim_correction = c("auto", "GG", "none")
)
```

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. Greenhouse–Geisser-corrected simulated p-values are available only for `"III"` and `"II"`.

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. G*Power's estimates can differ from `target_pes`, especially for small samples or terms with more degrees of freedom; a warning is issued when `gpower = TRUE`. The default `gpower = FALSE` is recommended.

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.

covariance — Optional within-subject covariance specification created by `within_covariance()`. Raw covariance matrices are not accepted, which avoids silently assuming a within-cell order. The default `NULL` uses standard deviations of `1` and a compound-symmetric correlation of `0.5` and issues a warning stating those defaults. A `within_covariance()` specification issues a warning when correlation pairs are omitted: its `default_correlation` applies only to those undefined pairs, while explicitly defined correlations are unchanged. All measurements use the specification's common marginal variance, while unequal correlations remain supported. For terms containing within-subject factors, the resolved covariance matrix is also used to derive a term-specific population Greenhouse–Geisser epsilon for `power_calc`. With the default `sim_correction = "auto"`, a population epsilon below `1 - 1e-8` also selects Greenhouse–Geisser-corrected simulated p-values for `ss_type` `"II"` or `"III"`.

means_pattern — Optional relative cell-mean shape created by `means_pattern()`. The sparse values are projected onto `term`, normalized, and uniformly rescaled to reach `target_pes`. If `NULL`, simulations use the package's deterministic linear/Kronecker pattern. For multi-df nonspherical within-subject terms, simulated power is conditional on this direction, so an explicit pattern is recommended when the expected shape is known.

`sim_correction` Sphericity correction for simulated p-values: `"auto"` (the default) uses Greenhouse–Geisser correction when the term-specific population epsilon is below `1 - 1e-8` and `ss_type` is `"II"` or `"III"`; `"GG"` requests correction for every simulated dataset; and `"none"` always uses the uncorrected univariate test. `"GG"` is an error with `ss_type = "I"`. For a between-only term or a within component with one degree of freedom, `"GG"` silently resolves to `"none"` because no sphericity correction applies.

Value

An `anovapowersim_curve` object. The `$results` tibble contains `n_per_cell`, `total_n`, `n_sims`, successful and failed simulation counts (`valid_sims`, `failed_sims`), the population nonsphericity correction (`epsilon`), numerator and denominator degrees of freedom (`num_df`, `den_df`), the noncentrality parameter (`ncp`), calculated power (`power_calc`), and simulated power (`power_sim`). The full-precision `power_sim` value, not its printed three-decimal representation, is used by adaptive searches. When `epsilon < 1`, the reported degrees of freedom and noncentrality are the corrected values used for `power_calc`. With the default `sim_correction = "auto"`, `power_sim` uses the Greenhouse–Geisser-corrected simulated p-value when `epsilon < 1 - 1e-8` and `ss_type` is `"III"` or `"II"`; otherwise it uses the uncorrected univariate test. Balanced simulation result objects also include `custom_means_pattern`, indicating whether the relative direction was supplied explicitly, plus `sim_correction` and `sim_correction_resolved` for the requested and applied simulated-test correction.

Simulated sphericity correction

`sim_correction` changes only `power_sim`. When Greenhouse–Geisser correction is selected, each simulated dataset is tested using its own sample-estimated epsilon from `car::Anova()`. `power_calc` is unchanged and always models the population-epsilon-adjusted test. Consequently, forcing `sim_correction = "GG"` under a truly spherical population can make `power_sim` slightly smaller than `power_calc`, because sample-epsilon GG correction is mildly conservative under sphericity.

Power is estimated for the prespecified corrected or uncorrected test. Conditional procedures that first run Mauchly's test and then decide whether to correct are not simulated.

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_curve(
  within = c(time = 4),
  term = "time",
  target_pes = 0.15,
  n_range = 30,
  means_pattern = means_pattern(
    time = 1, value = 0,
    time = 2, value = 0.3,
    time = 3, value = 0.5,
    time = 4, value = 0.6
  ),
  n_sims = 1000,
  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 searches upward from `n_start` until it brackets the target or reaches `n_max`. If `n_start` already reaches the target, the search probes the smallest sample size supported by the design to establish a lower bracket. It 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 = 5000,
```

```
  tol = 0.03,
  gpower = FALSE,
  progress = interactive(),
  parallel = FALSE,
  cores = NULL,
  seed = NULL,
  covariance = NULL,
  means_pattern = NULL,
  sim_correction = c("auto", "GG", "none")
)
```

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. Greenhouse–Geisser-corrected simulated p-values are available only for `"III"` and `"II"`.
n_start	Starting sample size per between-subject cell, not a lower bound for the search. If `NULL`, an initial value is estimated from calculated power and constrained to values that 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. G*Power's estimates can differ from `target_pes`, especially for small samples or terms with more degrees of freedom; a warning is issued when `gpower = TRUE`. The default `gpower = FALSE` is recommended.
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.

covariance — Optional within-subject covariance specification created by `within_covariance()`. Raw covariance matrices are not accepted, which avoids silently assuming a within-cell order. The default `NULL` uses standard deviations of `1` and a compound-symmetric correlation of `0.5` and issues a warning stating those defaults. A `within_covariance()` specification issues a warning when correlation pairs are omitted: its `default_correlation` applies only to those undefined pairs, while explicitly defined correlations are unchanged. All measurements use the specification's common marginal variance, while unequal correlations remain supported. For terms containing within-subject factors, the resolved covariance matrix is also used to derive a term-specific population Greenhouse–Geisser epsilon for `power_calc`. With the default `sim_correction = "auto"`, a population epsilon below `1 - 1e-8` also selects Greenhouse–Geisser-corrected simulated p-values for `ss_type "II"` or `"III"`.

means_pattern — Optional relative cell-mean shape created by `means_pattern()`. The sparse values are projected onto `term`, normalized, and uniformly rescaled to reach `target_pes`. If `NULL`, simulations use the package's deterministic linear/Kronecker pattern. For multi-df nonspherical within-subject terms, simulated power is conditional on this direction, so an explicit pattern is recommended when the expected shape is known.

sim_correction — Sphericity correction for simulated p-values: `"auto"` (the default) uses Greenhouse–Geisser correction when the term-specific population epsilon is below `1 - 1e-8` and `ss_type` is `"II"` or `"III"`; `"GG"` requests correction for every simulated dataset; and `"none"` always uses the uncorrected univariate test. `"GG"` is an error with `ss_type = "I"`. For a between-only term or a within component with one degree of freedom, `"GG"` silently resolves to `"none"` because no sphericity correction applies.

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
)
```

---

power_n_calc *Calculate the sample size needed for target ANOVA power*

Description

Calculation-only search for the per-between-cell sample size needed to reach a requested power for a balanced factorial ANOVA design. Unlike power_n(), this function does not run simulations, fit ANOVA models, or call car; numerator degrees of freedom, denominator degrees of freedom, noncentrality, and calculated power are obtained directly from the balanced design.

Usage

```
power_n_calc(
  between = NULL,
  within = NULL,
  term,
  target_pes,
  power = 0.9,
  alpha = 0.05,
  n_start = NULL,
  n_max = 5000,
  gpower = FALSE,
  epsilon = 1
)
```

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.
alpha	Significance threshold.
n_start	Starting sample size per between-subject cell, not a lower bound for the search. If NULL, starts from the smallest value with valid calculated-power degrees of freedom.
n_max	Maximum sample size per between-subject cell.

gpower — Logical; if TRUE, use the G*Power-style noncentrality convention* lambda = total_n * f^2. *The default* FALSE *uses* lambda = den_df * f^2. *G*Power's estimates can differ from target_pes, especially for small samples or terms with more degrees of freedom; a warning is issued when gpower = TRUE. The default gpower = FALSE is recommended.

epsilon — Population nonsphericity correction for the within-subject component of term. Must lie between the theoretical lower bound 1 / within_term_df and 1. The default 1 assumes sphericity. Values below 1 multiply the numerator degrees of freedom, denominator degrees of freedom, and noncentrality parameter. Nonsphericity corrections do not apply to purely between-subject terms.

Value

An anovapowersim_curve object with n_needed and total_n_needed. The $results tibble contains n_per_cell, total_n, n_sims, valid_sims, failed_sims, numerator and denominator degrees of freedom (num_df, den_df), the nonsphericity correction (epsilon), the noncentrality parameter (ncp), calculated power (power_calc), and simulated power (power_sim). For power_n_calc(), the simulation-specific columns are always NA. When epsilon < 1, num_df and den_df are the corrected degrees of freedom used in the power calculation.

Lifecycle

Experimental

power_n_calc() is experimental while the calculated-power search API and reporting format are refined.

Examples

```
power_n_calc(
  between = c(cond = 2),
  within = c(stim = 4),
  term = "cond:stim",
  target_pes = 0.14,
  power = 0.90,
  epsilon = 0.70
)
```

---

power_sensitivity — *Estimate ANOVA effect-size sensitivity at a fixed sample size*

Description

Searches for the minimum detectable partial eta squared at a fixed sample size and target power. Calculated power supplies an efficient initial estimate. Explicit simulations then bracket the target and refine the effect-size bracket using interpolation with midpoint fallback.

Usage

```
power_sensitivity(
  between = NULL,
  within = NULL,
  term,
  n,
  power = 0.9,
  n_sims = 10000,
  alpha = 0.05,
  ss_type = "III",
  pes_min = 1e-06,
  pes_max = 0.99,
  pes_tol = 0.001,
  gpower = FALSE,
  progress = interactive(),
  parallel = FALSE,
  cores = NULL,
  seed = NULL,
  covariance = NULL,
  means_pattern = NULL,
  sim_correction = c("auto", "GG", "none")
)
```

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.
`n`	Number of subjects per between-subject cell. For a purely within-subject design, this is the total sample size.
`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. Greenhouse–Geisser-corrected simulated p-values are available only for `"III"` and `"II"`.
`pes_min`	Lower bound of the partial eta-squared search interval.
`pes_max`	Upper bound of the partial eta-squared search interval.
`pes_tol`	Maximum width of the final simulated partial eta-squared bracket.

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. G*Power's estimates can differ from target_pes, especially for small samples or terms with more degrees of freedom; a warning is issued when gpower = TRUE. The default gpower = FALSE is recommended.

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.

covariance — Optional within-subject covariance specification created by within_covariance(). Raw covariance matrices are not accepted, which avoids silently assuming a within-cell order. The default NULL uses standard deviations of 1 and a compound-symmetric correlation of 0.5 and issues a warning stating those defaults. A within_covariance() specification issues a warning when correlation pairs are omitted: its default_correlation applies only to those undefined pairs, while explicitly defined correlations are unchanged. All measurements use the specification's common marginal variance, while unequal correlations remain supported. For terms containing within-subject factors, the resolved covariance matrix is also used to derive a term-specific population Greenhouse–Geisser epsilon for power_calc. With the default sim_correction = "auto", a population epsilon below 1 - 1e-8 also selects Greenhouse–Geisser-corrected simulated p-values for ss_type "II" or "III".

means_pattern — Optional relative cell-mean shape created by means_pattern(). The sparse values are projected onto term, normalized, and uniformly rescaled to reach target_pes. If NULL, simulations use the package's deterministic linear/Kronecker pattern. For multi-df nonspherical within-subject terms, simulated power is conditional on this direction, so an explicit pattern is recommended when the expected shape is known.

sim_correction — Sphericity correction for simulated p-values: "auto" (the default) uses Greenhouse–Geisser correction when the term-specific population epsilon is below 1 - 1e-8 and ss_type is "II" or "III"; "GG" requests correction for every simulated dataset; and "none" always uses the uncorrected univariate test. "GG" is an error with ss_type = "I". For a between-only term or a within component with one degree of freedom, "GG" silently resolves to "none" because no sphericity correction applies.

Value

An anovapowersim_sensitivity object. $pes_needed is the explicitly simulated upper effect-size bracket, or NA when pes_max does not achieve target power. $results contains every explicitly simulated effect size and its standard power diagnostics.

Lifecycle

Experimental

power_sensitivity() is experimental and is available only in the development version of anovapowersim. Its API and reporting format may change.

Examples

```
power_sensitivity(
  between = c(group = 2),
  within = c(time = 2),
  term = "group:time",
  n = 20,
  power = 0.90,
  n_sims = 100,
  pes_tol = 0.01,
  seed = 123
)
```

---

power_sensitivity_calc

Calculate ANOVA effect-size sensitivity at a fixed sample size

---

Description

Calculation-only counterpart to power_sensitivity(). The function searches for the minimum partial eta squared that reaches target power using calculated noncentral-F power, without simulating data or fitting ANOVA models.

Usage

```
power_sensitivity_calc(
  between = NULL,
  within = NULL,
  term,
  n,
  power = 0.9,
  alpha = 0.05,
  pes_min = 1e-06,
  pes_max = 0.99,
  pes_tol = 0.001,
  gpower = FALSE,
  epsilon = 1
)
```

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.
n	Number of subjects per between-subject cell. For a purely within-subject design, this is the total sample size.
power	Desired target power.
alpha	Significance threshold.
pes_min	Lower bound of the partial eta-squared search interval.
pes_max	Upper bound of the partial eta-squared search interval.
pes_tol	Maximum width of the final calculated partial eta-squared bracket.
gpower	Logical; if `TRUE`, use the G*Power-style noncentrality convention* `lambda = total_n * f^2`. *The default* `FALSE` *uses* `lambda = den_df * f^2`. *G*Power's estimates can differ from `target_pes`, especially for small samples or terms with more degrees of freedom; a warning is issued when `gpower = TRUE`. The default `gpower = FALSE` is recommended.
epsilon	Population nonsphericity correction for the within-subject component of `term`. Must lie between the theoretical lower bound `1 / within_term_df` and `1`. The default `1` assumes sphericity. Values below `1` multiply the numerator degrees of freedom, denominator degrees of freedom, and noncentrality parameter. Nonsphericity corrections do not apply to purely between-subject terms.

Value

An `anovapowersim_sensitivity` object. `$pes_needed` is the calculated upper effect-size bracket, or `NA` when `pes_max` does not achieve target power. `$results` contains every effect size evaluated by the calculated-power search; simulation-specific result columns are always `NA`.

Lifecycle

Experimental

`power_sensitivity_calc()` is experimental and is available only in the development version of `anovapowersim`. Its API and reporting format may change.

Examples

```
power_sensitivity_calc(
  between = c(group = 2),
  within = c(time = 3),
  term = "group:time",
  n = 30,
  power = 0.90,
  pes_tol = 0.001,
  gpower = TRUE,
  epsilon = 0.80
)
```

power_unbalanced *Simulate power for a fixed unbalanced ANOVA design*

Description

Estimates achieved power for exact, potentially unequal cell sizes and user-supplied population means under one common standard deviation. This function is simulation-only: it does not calculate power from a noncentral F distribution and does not scale the supplied sample sizes. A warning is issued when the deterministic reference data imply essentially zero effect for the tested term, which often indicates a typo, a wrong `term`, or means that contain only other effects.

Usage

```
power_unbalanced(
  design,
  term,
  covariance = NULL,
  n_sims = 10000,
  alpha = 0.05,
  ss_type = "III",
  progress = interactive(),
  parallel = FALSE,
  cores = NULL,
  seed = NULL,
  sim_correction = c("auto", "GG", "none")
)
```

Arguments

`design`	A complete design table created by `cell_design()`. Within-subject factors are read from the design's `within` attribute (set via `cell_design()`'s `within` argument), not supplied here.
`term`	Character scalar naming the ANOVA term to test.
`covariance`	Optional common covariance specification created by `unbalanced_covariance()`. The default `NULL` uses `sd = 1` and a correlation of `0.5` between within-subject measurements and issues a warning stating those defaults. For purely between-subject designs, use this argument to change the common SD; correlation settings are not applicable. When an `unbalanced_covariance()` specification omits some correlation pairs, a warning states that `default_correlation` is used only for those undefined pairs. The resolved covariance determines the term-specific population Greenhouse–Geisser epsilon used by `sim_correction = "auto"`.
`n_sims`	Number of simulated datasets.
`alpha`	Significance threshold.

ss_type	Sums-of-squares type: "III", "II", or "I". For unequal-N designs, "I" uses sequential, order-dependent hypotheses; a warning reports the factor order inherited from cell_design().
progress	Logical; if TRUE, show a text progress bar.
parallel	Logical; if TRUE, run simulations in parallel.
cores	Optional positive integer number of parallel workers.
seed	Optional integer seed for reproducibility.
sim_correction	Sphericity correction for simulated p-values: "auto" (the default) uses Greenhouse–Geisser correction when the term-specific population epsilon is below 1 - 1e-8 and ss_type is "II" or "III"; "GG" requests correction for every simulated dataset; and "none" always uses the uncorrected univariate test. "GG" is an error with ss_type = "I". For a between-only term or a within component with one degree of freedom, "GG" silently resolves to "none" because no sphericity correction applies.

Value

An anovapowersim_unbalanced_power object. $power and $achieved_power contain simulated power. $partial_eta_squared is the term effect size in a deterministic reference dataset. $epsilon is the population Greenhouse–Geisser epsilon for the tested term. $results also reports the common SD, simulated partial eta-squared distribution, and failed fits. Sample partial eta squared is upward-biased in finite samples, so mean_pes_sim, median_pes_sim, and the simulated interval are sampling diagnostics rather than estimates of the supplied population effect; use $partial_eta_squared as the reference effect. The object stores the requested sim_correction and applied sim_correction_resolved values.

Lifecycle

Experimental

power_unbalanced() is experimental and is available only in the development version of anovapowersim. Its API and reporting format may change.

Simulated sphericity correction

sim_correction governs only the simulated test. With "GG", each dataset uses its own sample-estimated Greenhouse–Geisser epsilon from car::Anova(). Under a truly spherical population, that sample correction is mildly conservative. Power is estimated for the prespecified corrected or uncorrected test; conditional Mauchly-then-correct procedures are not simulated. Unlike the balanced simulation functions, power_unbalanced() does not report a power_calc diagnostic.

Examples

```
design <- cell_design(
  group = "control", time = "pre",  n = 12, m = 10,
  group = "control", time = "post", n = 12, m = 11,
  group = "treatment", time = "pre",  n = 18, m = 10,
  group = "treatment", time = "post", n = 18, m = 13,
  within = "time"
```

```
)
power_unbalanced(
  design = design,
  term = "group:time",
  covariance = unbalanced_covariance(
    sd = 2,
    correlations = c("pre:post" = 0.7)
  ),
  n_sims = 100,
  seed = 123
)
```

`print.anovapowersim_achieved_power`

Print a fixed-sample achieved-power result

Description

Print a fixed-sample achieved-power result

Usage

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

Arguments

`x`	An `anovapowersim_achieved_power` object.
`...`	Unused.

Value

Invisibly returns `x`.

`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.

---

print.anovapowersim_sensitivity

Print a fixed-sample sensitivity result

Description

Print a fixed-sample sensitivity result

Usage

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

Arguments

x	An anovapowersim_sensitivity object.
...	Unused.

Value

Invisibly returns x.

print.anovapowersim_unbalanced_power

Print simulated power for an unbalanced design

Description

Print simulated power for an unbalanced design

Usage

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

Arguments

x	An anovapowersim_unbalanced_power 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`.

Covariance limitation

This manual helper does not accept `within_covariance()` specifications. It always simulates from the compound-symmetric covariance defined by `sd` and `r`. It therefore cannot reproduce a balanced power-function call that uses a custom `covariance`; use the power functions directly for that workflow.

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_achieved_power

Summarise a fixed-sample achieved-power result

Description

Summarise a fixed-sample achieved-power result

Usage

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

Arguments

`object`	An `anovapowersim_achieved_power` object.
`...`	Unused.

Value

A list with `header` and `results`, invisibly; printed to the console as well.

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.

summary.anovapowersim_sensitivity

Summarise a fixed-sample sensitivity result

Description

Summarise a fixed-sample sensitivity result

Usage

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

Arguments

`object`	An `anovapowersim_sensitivity` object.
`...`	Unused.

Value

A list with `header` and `results`, invisibly; printed to the console as well.

summary.anovapowersim_unbalanced_power *Summarise simulated power for an unbalanced design*

Description

Summarise simulated power for an unbalanced design

Usage

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

Arguments

`object`	An `anovapowersim_unbalanced_power` object.
`...`	Unused.

Value

A list with `header`, `design`, and `results`, invisibly; printed to the console as well.

unbalanced_covariance *Specify covariance for a means-based unbalanced design*

Description

Defines the common marginal standard deviation and within-subject correlation structure used by `power_unbalanced()`. The resulting covariance is shared by every between-subject cell.

Usage

```
unbalanced_covariance(sd = 1, default_correlation = 0.5, correlations = NULL)
```

Arguments

`sd`	Common positive finite marginal standard deviation. If omitted, `1` is used and a warning is issued.
`default_correlation`	Correlation in `(-1, 1)` used for unlisted pairs. When the specification is resolved for a design, a warning identifies how many pairs were not defined and makes clear that this default applies only to those pairs.
`correlations`	Optional named numeric vector of pair-specific correlations. Name pairs as `"cell1:cell2"`; pair order does not matter. For multiple within factors, cell names join their level values with `_`. Constructed names must be unique, and level values must not contain `:` because it separates the two cells in a pair name.

Value

An anovapowersim_unbalanced_covariance_spec object.

Lifecycle

Experimental

unbalanced_covariance() is experimental and is available only in the development version of anovapowersim. Its API may change.

Examples

```
unbalanced_covariance(
  sd = 2,
  default_correlation = 0.5,
  correlations = c("pre:post" = 0.7)
)
```

---

within_covariance *Specify a within-subject covariance structure*

Description

Creates a readable covariance specification for power_n() and power_curve(). Supply one common standard deviation and default correlation, then override only the individual correlation pairs that differ. Measurement names are generated from the within design; for example, within = c(time = 2, condition = 2) creates time1_condition1, time1_condition2, time2_condition1, and time2_condition2.

Usage

```
within_covariance(sd = 1, default_correlation = 0.5, correlations = NULL)
```

Arguments

sd	Common positive finite standard deviation for every measurement. If omitted, 1 is used and a warning is issued.
default_correlation	Finite correlation in (-1, 1) used for pairs not listed in correlations. When the specification is resolved for a design, a warning identifies how many pairs were not defined and makes clear that this default applies only to those pairs.
correlations	Optional named numeric vector of pair-specific correlations. Name each pair as "cell1:cell2", for example "time1:time2" = 0.7. Pair order does not matter.

Value

An anovapowersim_covariance_spec object. power_n() and power_curve() resolve it against their within design and construct the covariance matrix as sd^2 * R, where R is the correlation matrix. They also derive the tested term's population Greenhouse–Geisser epsilon from the resolved matrix and apply it to power_calc.

Examples

```
covariance <- within_covariance(
  sd = 1,
  default_correlation = 0.5,
  correlations = c(
    "time1_condition1:time1_condition2" = 0.6,
    "time2_condition1:time2_condition2" = 0.7
  )
)


power_n(
  within = c(time = 3, condition = 2),
  term = "time:condition",
  target_pes = 0.14,
  power = 0.90,
  n_sims = 100,
  covariance = covariance,
  seed = 123
)
```

Index

balanced_anova_design, 2
balanced_anova_design(), *6*, *30*

cell_design, 3
cell_design(), *8*, *26*, *27*
compute_scale_factor, 4
compute_scale_factor(), *7*

design_term_means, 5
design_term_means(), *2*, *5*, *30*

f_to_pes, 7

means_pattern, 8
means_pattern(), *4*, *6*, *11*, *15*, *19*, *23*

plot_power_curve, 9
power_achieved, 10
power_achieved(), *6*, *12*
power_achieved_calc, 12
power_curve, 14
power_curve(), *2*, *5*, *6*, *9*, *34*, *35*
power_n, 17
power_n(), *6*, *20*, *34*, *35*
power_n_calc, 20
power_n_calc(), *7*
power_sensitivity, 21
power_sensitivity(), *24*
power_sensitivity_calc, 24
power_unbalanced, 26
power_unbalanced(), *3*, *33*
print.anovapowersim_achieved_power, 28
print.anovapowersim_curve, 28
print.anovapowersim_sensitivity, 29
print.anovapowersim_unbalanced_power, 30

simulate_design_dataset, 30
simulate_design_dataset(), *2*
summary.anovapowersim_achieved_power, 31
summary.anovapowersim_curve, 32
summary.anovapowersim_sensitivity, 32
summary.anovapowersim_unbalanced_power, 33

unbalanced_covariance, 33
unbalanced_covariance(), *3*, *26*

within_covariance, 34
within_covariance(), *6*, *11*, *15*, *19*, *23*, *31*