Package ‘Superpower’

June 24, 2025

Title Simulation-Based Power Analysis for Factorial Designs

Version 0.2.4

Description Functions to perform simulations of ANOVA designs of up to three factors. Calculates the observed power and average observed effect size for all main effects and interactions in the ANOVA, and all simple comparisons between conditions. Includes functions for analytic power calculations and additional helper functions that compute effect sizes for ANOVA designs, observed error rates in the simulations, and functions to plot power curves. Please see Lakens, D., & Caldwell, A. R. (2021). ``Simulation-Based Power Analysis for Factorial Analysis of Variance Designs''. <doi:10.1177/2515245920951503>.

URL https://aaroncaldwell.us/SuperpowerBook/

BugReports https://github.com/arcaldwell49/Superpower/issues

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports MASS, afex, emmeans, ggplot2, reshape2, stats, dplyr, magrittr, tidyselect, tidyr

Suggests knitr, rmarkdown, pwr, testthat, covr, jmvcore, spelling

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Aaron Caldwell [aut, cre],
Daniel Lakens [aut],
Lisa DeBruine [ctb],
Jonathon Love [ctb],
Frederik Aust [ctb] (ORCID: <https://orcid.org/0000-0003-4900-788X>)

Maintainer Aaron Caldwell <arcaldwell49@gmail.com>

Repository CRAN

Date/Publication 2025-06-24 19:30:02 UTC

Contents

`alpha_standardized` *Compute standardized alpha level based on unstandardized alpha level and the number of observations N.*

Description

Compute standardized alpha level based on unstandardized alpha level and the number of observations N.

Usage

```
alpha_standardized(alpha, N, standardize_N = 100)
```

Arguments

`alpha`	The unstandardized alpha level (e.g., 0.05), independent of the sample size.
`N`	The number of observations (e.g., the sample size) in the dataset
`standardize_N`	The number of observations (e.g., the sample size) you want to use to standardize the alpha level for. Defaults to 100 (base on Good, 1982).

References

Good, I. J. (1982). C140. Standardized tail-area probabilities. Journal of Statistical Computation and Simulation, 16(1), 65–66. <https://doi.org/10.1080/00949658208810607>

Examples

```
## Check it yields .05 for N = 100:
alpha_standardized(alpha = 0.05, N = 100)
## Check it yields .05 for N = 200:
alpha_standardized(alpha = 0.07071068, N = 200)
## Which alpha should we use with N = 200?
alpha_standardized(alpha = 0.05, N = 200)
## You can change the standardization N, repeating the example above:
alpha_standardized(alpha = 0.05, N = 100, standardize_N = 200)
```

ANCOVA_analytic *Power Calculations for Factorial ANCOVAs*

Description

Complete power analyses for ANCOVA omnibus tests and contrasts. This function does not support within subjects factors.

Usage

```
ANCOVA_analytic(
  design,
  mu,
  n = NULL,
  sd,
  r2 = NULL,
  n_cov,
  alpha_level = Superpower_options("alpha_level"),
  beta_level = NULL,
  cmats = list(),
  label_list = NULL,
  design_result = NULL,
  round_up = TRUE
)
```

Arguments

design	Output from the ANOVA_design function
mu	Vector specifying mean for each condition
n	Sample size in each condition
sd	Standard deviation for all conditions (or a vector specifying the sd for each condition)

r2	Coefficient of Determination of the model with only the covariates
n_cov	Number of covariates
alpha_level	Alpha level used to determine statistical significance
beta_level	Type II error probability (power/100-1)
cmats	List of matrices for specific contrasts of interest
label_list	An optional list to specify the factor names and condition (recommended, if not used factors and levels are indicated by letters and numbers).
design_result	Output from the ANOVA_design function
round_up	Logical indicator (default = TRUE) for whether to round up sample size calculations to nearest whole number

Value

One, or two, data frames containing the power analysis results from the power analysis for the omnibus ANCOVA (main_results) or contrast tests (contrast_results). In addition, every F-test (aov_list and con_list) is included in a list of power.htest results. Lastly, a (design_param) list containing the design parameters is also included in the results.

References

Shieh, G. (2020). Power analysis and sample size planning in ANCOVA designs. Psychometrika, 85(1), 101-120.

Examples

```
# Simple 2x3 ANCOVA

ANCOVA_analytic(
design = "2b*3b",
mu = c(400, 450, 500,
      400, 500, 600),
n_cov = 3,
sd = 100,
r2 = .25,
alpha_level = .05,
beta_level = .2,
round_up = TRUE
)
```

ANCOVA_contrast *Power Calculations for ANCOVA Contrasts*

Description

Complete power analyses for specific ANCOVA contrasts. This function does not support within subjects factors.

Usage

```
ANCOVA_contrast(
  cmat,
  mu,
  n = NULL,
  sd,
  r2 = NULL,
  n_cov,
  alpha_level = Superpower_options("alpha_level"),
  beta_level = NULL,
  round_up = TRUE
)
```

Arguments

cmat	Matrix of the specific contrasts of interest
mu	Vector specifying mean for each condition
n	Sample size in each condition
sd	Standard deviation for all conditions (or a vector specifying the sd for each condition)
r2	Coefficient of Determination of the model with only the covariates
n_cov	Number of covariates
alpha_level	Alpha level used to determine statistical significance
beta_level	Type II error probability (power/100-1)
round_up	Logical indicator (default = TRUE) for whether to round up sample size calculations to nearest whole number

Value

Object of class "power.htest", a list of the arguments (including the computed one) augmented with method and note elements.

References

Shieh, G. (2020). Power analysis and sample size planning in ANCOVA designs. Psychometrika, 85(1), 101-120.

Examples

```
ANCOVA_contrast(cmat = c(-1,1),
n = 15,
mu = c(0,1),
sd = 1,
r2 = .2,
n_cov = 1)
```

ancova_power-methods *Methods for ancova_power objects*

Description

Methods defined for objects returned from the ANCOVA_analytic function.

Usage

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

## S3 method for class 'ancova_power'
plot(x, ...)
```

Arguments

x	object of class ancova_power as returned from one of the simulation functions in Superpower.
...	further arguments passed through, see description of return value

Value

print Prints short summary of the simulation result

plot Returns a meansplot of from the defined design

ANOVA_compromise *Justify your alpha level by minimizing or balancing Type 1 and Type 2 error rates for ANOVAs.*

Description

Justify your alpha level by minimizing or balancing Type 1 and Type 2 error rates for ANOVAs.

Usage

```
ANOVA_compromise(
  design_result,
  correction = Superpower_options("correction"),
  emm = Superpower_options("emm"),
  emm_model = Superpower_options("emm_model"),
  contrast_type = Superpower_options("contrast_type"),
  emm_comp,
  costT1T2 = 1,
  priorH1H0 = 1,
```

```
  error = c("minimal", "balance"),
  liberal_lambda = Superpower_options("liberal_lambda")
)
```

Arguments

design_result	Output from the ANOVA_design function
correction	Set a correction of violations of sphericity. This can be set to "none", "GG" Greenhouse-Geisser, and "HF" Huynh-Feldt
emm	Set to FALSE to not perform analysis of estimated marginal means
emm_model	Set model type ("multivariate", or "univariate") for estimated marginal means
contrast_type	Select the type of comparison for the estimated marginal means. Default is pairwise. See ?emmeans::'contrast-methods' for more details on acceptable methods.
emm_comp	Set the comparisons for estimated marginal means comparisons. This is a factor name (a), combination of factor names (a+b), or for simple effects a | sign is needed (a|b)
costT1T2	Relative cost of Type 1 errors vs. Type 2 errors.
priorH1H0	How much more likely a-priori is H1 than H0? Default is 1: equally likely.
error	Either "minimal" to minimize error rates, or "balance" to balance error rates.
liberal_lambda	Logical indicator of whether to use the liberal (cohen_f^2*(num_df+den_df)) or conservative (cohen_f^2*den_df) calculation of the noncentrality (lambda) parameter estimate. Default is FALSE.

Value

Returns dataframe with simulation data (power and effect sizes!), optimal alpha level, obtained beta error rate (1-power/100), and objective (see below for details). If NA is obtained in a alpha/beta/objective columns this indicates there is no effect for this particular comparison. Also returns alpha-beta compromise plots for all comparisons. Note: Cohen's f = sqrt(pes/1-pes) and the noncentrality parameter is = f^2*df(error)

"aov_comp" A dataframe of ANOVA-level results.

"aov_plotlist" List of plots for ANOVA-level effects

"manova_comp" A dataframe of MANOVA-level results.

"manova_plotlist" List of plots for MANOVA-level effects.

"emmeans_comp" A dataframe of ANOVA-level results.

"emm_plotlist" List of plots for estimated marginal means contrasts.

alpha = alpha or Type 1 error that minimizes or balances combined error rates beta = beta or Type 2 error that minimizes or balances combined error rates objective = value that is the result of the minimization, either 0 (for balance) or the combined weighted error rates

References

too be added

Examples

```
## Not run:
design_result <- ANOVA_design(design = "3b*2w",
n = 6,
mu = c(1, 2, 2, 3, 3, 4),
sd = 3,
plot = FALSE)
example = ANOVA_compromise(design_result,emm = TRUE,emm_comp = "a")

## End(Not run)
```

ANOVA_design — *Design function used to specify the parameters to be used in simulations*

Description

Design function used to specify the parameters to be used in simulations

Usage

```
ANOVA_design(
  design,
  n,
  mu,
  sd,
  r = 0,
  label_list = NULL,
  labelnames = NULL,
  plot = Superpower_options("plot")
)
```

Arguments

design	String specifying the ANOVA design.
n	Sample size in each condition
mu	Vector specifying mean for each condition
sd	standard deviation for all conditions (or a vector specifying the sd for each condition)
r	Correlation between dependent variables (single value or matrix)
label_list	An optional list to specify the factor names and condition (recommended, if not used factors and levels are indicated by letters and numbers).
labelnames	Optional vector to specifying factor and condition names. This parameter is deprecated and will be overridden by input from label_list.
plot	Should means plot be printed (defaults to TRUE)

Value

Returns single list with simulated data, design, design list, factor names, formulas for ANOVA, means, sd, correlation, sample size per condition, correlation matrix, covariance matrix, design string, labelnames, labelnameslist, factor names, meansplot

"dataframe" A sample dataframe of what data could look like given the proposed parameters.

"design" aov The design string, e.g. "2b*2w".

"design_list" The list of variables in the design.

"frml1" The first formula created for this design.

"frml2" The second formula created for this design.

"mu" Vector of means.

"sd" Vector of standard deviations.

"r" Common correlation coefficient.

"n" Sample size per cell. Can be entered as a single value or list of sample sizes for each condition. If unequal n is entered then the design can only be passed onto ANOVA_power.

"cor_mat" The correlation matrix.

"sigmatrix" The variance-covariance matrix.

"design_factors" Total number of within-subjects factors.

"labelnames" List of the label names.

"labelnameslist" Secondary list of labelnames

"factornames" List of the factor titles.

"meansplot" Plot of the experimental design.

Warnings

Varying the sd or r (e.g., entering multiple values) violates assumptions of homoscedascity and sphericity respectively

Examples

```
## Set up a within design with 2 factors, each with 2 levels,
## with correlation between observations of 0.8,
## 40 participants (who do all conditions), and standard deviation of 2
## with a mean pattern of 1, 0, 1, 0, conditions labeled 'condition' and
## 'voice', with names for levels of "cheerful", "sad", and "human", "robot"
ANOVA_design(design = "2w*2w", n = 40, mu = c(1, 0, 1, 0), sd = 2, r = 0.8,
      label_list= list(condition = c("cheerful", "sad"),
      voice = c("human", "robot")))
```

ANOVA_exact *Simulates an exact dataset (mu, sd, and r represent empirical, not population, mean and covariance matrix) from the design to calculate power*

Description

Simulates an exact dataset (mu, sd, and r represent empirical, not population, mean and covariance matrix) from the design to calculate power

Usage

```
ANOVA_exact(
  design_result,
  correction = Superpower_options("correction"),
  alpha_level = Superpower_options("alpha_level"),
  verbose = Superpower_options("verbose"),
  emm = Superpower_options("emm"),
  emm_model = Superpower_options("emm_model"),
  contrast_type = Superpower_options("contrast_type"),
  liberal_lambda = Superpower_options("liberal_lambda"),
  emm_comp
)

ANOVA_exact2(
  design_result,
  correction = Superpower_options("correction"),
  alpha_level = Superpower_options("alpha_level"),
  verbose = Superpower_options("verbose"),
  emm = Superpower_options("emm"),
  emm_model = Superpower_options("emm_model"),
  contrast_type = Superpower_options("contrast_type"),
  emm_comp,
  liberal_lambda = Superpower_options("liberal_lambda")
)
```

Arguments

design_result	Output from the ANOVA_design function
correction	Set a correction of violations of sphericity. This can be set to "none", "GG" Greenhouse-Geisser, and "HF" Huynh-Feldt
alpha_level	Alpha level used to determine statistical significance
verbose	Set to FALSE to not print results (default = TRUE)
emm	Set to FALSE to not perform analysis of estimated marginal means
emm_model	Set model type ("multivariate", or "univariate") for estimated marginal means

contrast_type Select the type of comparison for the estimated marginal means. Default is pairwise. See the emmeans package on "contrast-methods" for more details on acceptable methods.

liberal_lambda Logical indicator of whether to use the liberal (cohen_f^2*(num_df+den_df)) or conservative (cohen_f^2*den_df) calculation of the noncentrality (lambda) parameter estimate. Default is FALSE.

emm_comp Set the comparisons for estimated marginal means comparisons. This is a factor name (a), combination of factor names (a+b), or for simple effects a | sign is needed (a|b)

Value

Returns dataframe with simulation data (power and effect sizes!), anova results and simple effect results, plot of exact data, and alpha_level. Note: Cohen's f = sqrt(pes/1-pes) and the noncentrality parameter is = f^2*df(error)

"dataframe" A dataframe of the simulation result.

"aov_result" aov object returned from aov_car.

"main_result" The power analysis results for ANOVA level effects.

"pc_results" The power analysis results for the pairwise (t-test) comparisons.

"emm_results" The power analysis results of the pairwise comparison results.

"manova_results" Default is "NULL". If a within-subjects factor is included, then the power of the multivariate (i.e. MANOVA) analyses will be provided.

"alpha_level" The alpha level, significance cut-off, used for the power analysis.

"method" Record of the function used to produce the simulation

"plot" A plot of the dataframe from the simulation; should closely match the meansplot in ANOVA_design

Functions

- ANOVA_exact2(): An extension of ANOVA_exact that uses the effect sizes calculated from very large sample size empirical simulation. This allows for small sample sizes, where ANOVA_exact cannot, while still accurately estimating power. However, model objects (emmeans and aov) are not included as output, and pairwise (t-test) results are not currently supported.

Warnings

Varying the sd or r (e.g., entering multiple values) violates assumptions of homoscedascity and sphericity respectively

Examples

```
## Set up a within design with 2 factors, each with 2 levels,
## with correlation between observations of 0.8,
## 40 participants (who do all conditions), and standard deviation of 2
## with a mean pattern of 1, 0, 1, 0, conditions labeled 'condition' and
## 'voice', with names for levels of "cheerful", "sad", amd "human", "robot"
design_result <- ANOVA_design(design = "2w*2w", n = 40, mu = c(1, 0, 1, 0),
```

```
      sd = 2, r = 0.8, labelnames = c("condition", "cheerful",
      "sad", "voice", "human", "robot"))
exact_result <- ANOVA_exact(design_result, alpha_level = 0.05)
```

ANOVA_power *Simulation function used to estimate power*

Description

Simulation function used to estimate power

Usage

```
ANOVA_power(
  design_result,
  alpha_level = Superpower_options("alpha_level"),
  correction = Superpower_options("correction"),
  p_adjust = "none",
  nsims = 1000,
  seed = NULL,
  verbose = Superpower_options("verbose"),
  emm = Superpower_options("emm"),
  emm_model = Superpower_options("emm_model"),
  contrast_type = Superpower_options("contrast_type"),
  emm_p_adjust = "none",
  emm_comp = NULL
)
```

Arguments

design_result	Output from the ANOVA_design function
alpha_level	Alpha level used to determine statistical significance
correction	Set a correction of violations of sphericity. This can be set to "none", "GG" Greenhouse-Geisser, and "HF" Huynh-Feldt
p_adjust	Correction for multiple comparisons. This will adjust p values for ANOVA/MANOVA level effects; see ?p.adjust for options
nsims	number of simulations to perform
seed	Set seed for reproducible results
verbose	Set to FALSE to not print results (default = TRUE)
emm	Set to FALSE to not perform analysis of estimated marginal means
emm_model	Set model type ("multivariate", or "univariate") for estimated marginal means
contrast_type	Select the type of comparison for the estimated marginal means. Default is pairwise. See ?emmeans::‘contrast-methods‘ for more details on acceptable methods.

`emm_p_adjust` Correction for multiple comparisons; default is "none". See ?summary.emmGrid for more details on acceptable methods.

`emm_comp` Set the comparisons for estimated marginal means comparisons. This is a factor name (a), combination of factor names (a+b), or for simple effects a | sign is needed (a|b)

Value

Returns dataframe with simulation data (p-values and effect sizes), anova results (type 3 sums of squares) and simple effect results, and plots of p-value distribution.

`"sim_data"` Output from every iteration of the simulation

`"main_result"` The power analysis results for ANOVA effects.

`"pc_results"` The power analysis results for pairwise comparisons.

`"manova_results"` Default is "NULL". If a within-subjects factor is included, then the power of the multivariate (i.e. MANOVA) analyses will be provided.

`"emm_results"` The power analysis results of the estimated marginal means.

`"plot1"` Distribution of p-values from the ANOVA results.

`"plot2"` Distribution of p-values from the pairwise comparisons results.

`"correction"` The correction for sphericity applied to the simulation results.

`"p_adjust"` The p-value adjustment applied to the simulation results for ANOVA/MANOVA omnibus tests and t-tests.

`"emm_p_adjust"` The p-value adjustment applied to the simulation results for the estimated marginal means.

`"nsims"` The number of simulations run.

`"alpha_level"` The alpha level, significance cut-off, used for the power analysis.

`"method"` Record of the function used to produce the simulation

References

too be added

Examples

```
## Not run:
## Set up a within design with 2 factors, each with 2 levels,
## with correlation between observations of 0.8,
## 40 participants (who do all conditions), and standard deviation of 2
## with a mean pattern of 1, 0, 1, 0, conditions labeled 'condition' and
## 'voice', with names for levels of "cheerful", "sad", amd "human", "robot"
design_result <- ANOVA_design(design = "2w*2w", n = 40, mu = c(1, 0, 1, 0),
      sd = 2, r = 0.8, labelnames = c("condition", "cheerful",
      "sad", "voice", "human", "robot"))
power_result <- ANOVA_power(design_result, alpha_level = 0.05,
      p_adjust = "none", seed = 2019, nsims = 10)

## End(Not run)
```

design_aov-methods *Methods for design_aov objects*

Description

Methods defined for objects returned from the ANOVA_design functions.

Usage

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

## S3 method for class 'design_aov'
plot(x, ...)
```

Arguments

x	object of class `design_aov` as returned from `ANOVA_design`
...	further arguments passed through, see description of return value for details. `ANOVA_design`.

Value

`print` Prints short summary of the study design created from `ANOVA_design` function

`plot` Returns `meansplot` from created from the `ANOVA_design` function

emmeans_power *Compute power for* **emmeans** *contrasts*

Description

Computes power based on t value and degrees of freedom for contrasts. *Do not use to calculate "observed power" for empirical datasets (Hoenig & Heisey, 2001).*

Usage

```
emmeans_power(x, ...)

## S3 method for class 'emmGrid'
emmeans_power(x, ...)

## S3 method for class 'summary_em'
emmeans_power(x, ...)

## S3 method for class 'data.frame'
```

```
emmeans_power(
  x,
  alpha_level = Superpower_options("alpha_level"),
  liberal_lambda = Superpower_options("liberal_lambda"),
  ...
)
```

Arguments

`x`	emmGrid-class. Grid of contrasts to estimate power from.
`...`	Other arguments passed to the function if object is not already a emmGrid object.
`alpha_level`	Alpha level used to determine statistical significance
`liberal_lambda`	Logical indicator of whether to use the liberal (cohen_f^2*(num_df+den_df)) or conservative (cohen_f^2*den_df) calculation of the noncentrality (lambda) parameter estimate. Default is FALSE.

Details

Note that calculation of power is based on the F- and t-ratio assuming two-sided testing. Thus, the function does not honor adjustments of the testing procedure due to either one-sided testing (or two-one sided tests) or corrections for multiple comparisons via the `p.adjust` option in emmeans.

Power for one-sided tests can be calculated, if the means of the simulated dataset are consistent with the directional hypothesis, by doubling `alpha_level`. Similarly, power for Bonferroni-corrected contrasts can be calculated by adjusting `alpha_level` accordingly (see examples). `...` Other arguments passed onto the function

Value

Returns dataframe with simulation data (power and effect sizes!), anova results and simple effect results, plot of exact data, and alpha_level. Note: Cohen's f = sqrt(pes/1-pes) and the noncentrality parameter is = f^2*df(error)

`"dataframe"` A dataframe of the simulation result.

`"aov_result"` aov object returned from aov_car.

`"main_result"` The power analysis results for ANOVA level effects.

`"pc_results"` The power analysis results for the pairwise (t-test) comparisons.

`"emm_results"` The power analysis results of the pairwise comparison results.

`"manova_results"` Default is "NULL". If a within-subjects factor is included, then the power of the multivariate (i.e. MANOVA) analyses will be provided.

`"alpha_level"` The alpha level, significance cut-off, used for the power analysis.

`"method"` Record of the function used to produce the simulation

`"plot"` A plot of the dataframe from the simulation; should closely match the meansplot in `ANOVA_design`

Author(s)

Frederik Aust

References

Hoenig, J. M., & Heisey, D. M. (2001). The Abuse of Power. The American Statistician, 55(1), 19–24. https://doi.org/10.1198/000313001300339897

Examples

```
## Not run:
# Set up a within design with 2 factors, each with 2 levels
design_result <- ANOVA_design(design = "2w*2w",
n = 40, mu = c(1, 0, 1, 0),
sd = 2, r = 0.8,
labelnames = c("condition", "cheerful",
 "sad", "voice", "human", "robot"))

exact_result <- ANOVA_exact(design_result,
alpha_level = 0.05, verbose = FALSE,
emm = TRUE, contrast_type = "pairwise")

# Power for pairwise contrasts
exact_result$emm_results

# Corresponding emmeans contrasts
exact_result$emmeans$contrasts

# Manually recalculate power
emmeans_power(exact_result$emmeans$contrasts,
alpha_level = 0.05)

# Calculate power for Bonferroni-adjusted pairwise comparisons
n_contrasts <- nrow(as.data.frame(exact_result$emmeans$contrasts))
emmeans_power(exact_result$emmeans$contrasts,
alpha_level = 0.05 / n_contrasts)

# Calculate power for one-sided custom contrasts
exact_result$emmeans$emmeans
custom_contrast <- contrast(exact_result$emmeans$emmeans,
list(robot_vs_sad_human = c(0, 1, -0.5, -0.5)))
emmeans_power(custom_contrast,
 alpha_level = 0.05 * 2)

# Calculate power for follow-up ANOVA
follow_up <- joint_tests(exact_result$emmeans$emmeans,
by = "condition")
emmeans_power(follow_up,
alpha_level = 0.05 / 2)
emmeans_power(emmeans(exact_result$emmeans$emmeans,
pairwise ~ voice | condition)$contrasts,
alpha_level = 0.05 / 2)

## End(Not run)
```

`morey_plot.ttest` *Plot out power sensitivity plots for t or F tests*

Description

Plot out power sensitivity plots for t or F tests

Usage

```
morey_plot.ttest(
  es = seq(0, 1, 0.05),
  n = NULL,
  type = c("two.sample", "one.sample", "paired"),
  alternative = c("two.sided", "one.sided"),
  alpha_level = Superpower_options("alpha_level")
)

morey_plot.ftest(
  es = seq(0, 1, 0.05),
  num_df = 1,
  den_df = NULL,
  alpha_level = Superpower_options("alpha_level"),
  liberal_lambda = Superpower_options("liberal_lambda")
)
```

Arguments

`es`	Effect size magnitudes to include on the plot; either cohen's f or cohen's d depending on whether it is an F-test or t-test
`n`	Sample size (t-test only) per group (two sample), total number of pairs (paired samples), or total observations (one-sample); only applies to t-test
`type`	string specifying the type of t test. Can be abbreviated. (t-test only)
`alternative`	one- or two-sided test. Can be abbreviated. (t-test only)
`alpha_level`	vector of alpha levels; default is 0.05
`num_df`	Numerator degrees of freedom for an F-test.
`den_df`	Denominator degrees of freedom for an F-test.
`liberal_lambda`	Logical indicator of whether to use the liberal (cohen_f^2*(num_df+den_df)) or conservative (cohen_f^2*den_df) calculation of the noncentrality (lambda) parameter estimate. Default is FALSE.

Value

Returns plots of effect size (x-axis)

Functions

- `morey_plot.ttest()`: Power-sensitivity plot for t-tests
- `morey_plot.ftest()`: Power-sensitivity plot for F-tests

References

Morey, R.D. (2020). Power and precision Why the push for replacing "power" with "precision" is misguided. Retrieved from: https://richarddmorey.medium.com/power-and-precision-47f644ddea5e

Examples

```
## Not run:
# t-test example ------
# Sensitivity for cohen's d from .1 to .5
# sample sizes of 10 and 20
# alpha levels .05 and .075
# type will be paired and one sided
# Set effect sizes with seq function (?seq)

morey_plot.ttest(es = seq(.1,.5,.01),
n = c(10,20),
alpha_level = c(.05,.075),
type = "paired",
alternative = "one.sided")

## End(Not run)
```

`mu_from_ES` *Convenience function to calculate the means for between designs with one factor (One-Way ANOVA). Can be used to determine the means that should yield a specified effect sizes (expressed in Cohen's f).*

Description

Convenience function to calculate the means for between designs with one factor (One-Way ANOVA). Can be used to determine the means that should yield a specified effect sizes (expressed in Cohen's f).

Usage

```
mu_from_ES(K, ES)
```

Arguments

K	Number of groups (2, 3, or 4)
ES	Effect size (eta-squared)

Value

Returns vector of means

References

Albers, C., & Lakens, D. (2018). When power analyses based on pilot data are biased: Inaccurate effect size estimators and follow-up bias. Journal of Experimental Social Psychology, 74, 187–195. https://doi.org/10.1016/j.jesp.2017.09.004

Examples

```
## Medium effect size (eta-squared), 2 groups
ES <- 0.0588
K <- 2
mu_from_ES(K = K, ES = ES)
```

---

optimal_alpha — *Justify your alpha level by minimizing or balancing Type 1 and Type 2 error rates.*

Description

Justify your alpha level by minimizing or balancing Type 1 and Type 2 error rates.

Usage

```
optimal_alpha(
  power_function,
  costT1T2 = 1,
  priorH1H0 = 1,
  error = c("minimal", "balance"),
  plot = Superpower_options("plot")
)
```

Arguments

power_function	Function that outputs the power, calculated with an analytic function.
costT1T2	Relative cost of Type 1 errors vs. Type 2 errors.
priorH1H0	How much more likely a-priori is H1 than H0?
error	Either "minimal" to minimize error rates, or "balance" to balance error rate
plot	When set to TRUE, automatically outputs a plot of alpha (x-axis) and beta (y-axis) error rates

Value

alpha = alpha or Type 1 error that minimizes or balances combined error rates beta = beta or Type 2 error that minimizes or balances combined error rates objective = value that is the result of the minimization, either 0 (for balance) or the combined weighted error rates plot =

References

too be added

Examples

```
## Optimize power for a independent t-test, smallest effect of interest
## d = 0.5, 100 participants per condition
res <- optimal_alpha(power_function = "pwr::pwr.t.test(d = 0.5, n = 100,
sig.level = x, type = 'two.sample', alternative = 'two.sided')$power")
res$alpha
res$beta
```

---

opt_alpha-methods *Methods for opt_alpha objects*

Description

Methods defined for objects returned from the optimal_alpha and ANOVA_compromise functions.

Usage

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

## S3 method for class 'opt_alpha'
plot(x, ...)
```

Arguments

x	object of class `opt_alpha` as returned from one of the optimal alpha functions in Superpower.
...	further arguments passed through, see description of return value for details. `ANOVA_compromise`.

Value

`print` Prints short summary of the optimal alpha results

`plot` Returns a plot

plot_power *Convenience function to plot power across a range of sample sizes.*

Description

Convenience function to plot power across a range of sample sizes.

Usage

```
plot_power(
  design_result,
  alpha_level = Superpower_options("alpha_level"),
  min_n = 7,
  max_n = 100,
  desired_power = 90,
  plot = Superpower_options("plot"),
  emm = Superpower_options("emm"),
  emm_model = Superpower_options("emm_model"),
  contrast_type = Superpower_options("contrast_type"),
  emm_comp,
  verbose = Superpower_options("verbose"),
  exact2 = FALSE,
  liberal_lambda = Superpower_options("liberal_lambda")
)
```

Arguments

`design_result`	Output from the ANOVA_design function
`alpha_level`	Alpha level used to determine statistical significance
`min_n`	Minimum sample size in power curve. Cannot be less than or equal to the product of factors. E.g., if design = "2b*2b" then min_n must be at least 5 (2*2+1=5)
`max_n`	Maximum sample size in power curve.
`desired_power`	Desired power (e.g., 80, 90). N per group will be highlighted to achieve this desired power in the plot. Defaults to 90.
`plot`	Should power plot be printed automatically (defaults to TRUE)
`emm`	Set to FALSE to not perform analysis of estimated marginal means
`emm_model`	Set model type ("multivariate", or "univariate") for estimated marginal means
`contrast_type`	Select the type of comparison for the estimated marginal means
`emm_comp`	Set the comparisons for estimated marginal means comparisons. This is a factor name (a), combination of factor names (a+b), or for simple effects a | sign is needed (a|b)
`verbose`	Set to FALSE to not print results (default = TRUE)

`exact2` Logical indicator for which `ANOVA_exact` function (`ANOVA_exact` or `ANOVA_exact2`) to use in the plots. Default is FALSE which uses `ANOVA_exact` which has sample size limitations.

`liberal_lambda` Logical indicator of whether to use the liberal (cohen_f^2*(num_df+den_df)) or conservative (cohen_f^2*den_df) calculation of the noncentrality (lambda) parameter estimate. Default is FALSE.

Value

Returns plot with power curves for the ANOVA, and a dataframe with the summary data.

`"plot_ANOVA"` Plot of power curves from ANOVA results.

`"plot_MANOVA"` Plot of power curves from MANOVA results. Returns NULL if no within-subject factors.

`"plot_emm"` Plot of power curves from MANOVA results. Returns NULL if emm = FALSE.

`"anova_n"` Achieved Power and Sample Size for ANOVA-level effects.

`"manova_n"` Achieved Power and Sample Size for MANOVA-level effects.

`"emm_n"` Achieved Power and Sample Size for estimated marginal means.

`"power_df"` The tabulated ANOVA power results.

`"power_df_manova"` The tabulated MANOVA power results. Returns NULL if no within-subject factors.

`"power_df_emm"` The tabulated Estimated Marginal Means power results. Returns NULL if emm = FALSE.

`"effect_sizes"` Effect sizes (partial eta-squared) from ANOVA results.

`"effect_sizes_manova"` Effect sizes (Pillai's Trace) from MANOVA results. Returns NULL if no within-subject factors.

`"effect_sizes_emm"` Effect sizes (cohen's f) estimated marginal means results. Returns NULL if emm = FALSE.

References

too be added

Examples

```
## Not run:
design_result <- ANOVA_design(design = "3b",
                              n = 20,
                              mu = c(0,0,0.3),
                              sd = 1,
                              labelnames = c("condition",
                              "cheerful", "neutral", "sad"))

plot_power(design_result, min_n = 50, max_n = 70, desired_power = 90)

## End(Not run)
```

`power.ftest` *Power Calculations for an F-test*

Description

Compute power of test or determine parameters to obtain target power. Inspired by the pwr.f2.test function in the pwr package, but allows for varying noncentrality parameter estimates for a more liberal (default in pwr.f2.test) or conservative (default in this function) estimates (see Aberson, Chapter 5, pg 72).

Usage

```
power.ftest(
  num_df = NULL,
  den_df = NULL,
  cohen_f = NULL,
  alpha_level = Superpower_options("alpha_level"),
  beta_level = NULL,
  liberal_lambda = Superpower_options("liberal_lambda")
)
```

Arguments

`num_df`	degrees of freedom for numerator
`den_df`	degrees of freedom for denominator
`cohen_f`	Cohen's f effect size. Note: this is the sqrt(f2) if you are used to using pwr.f2.test
`alpha_level`	Alpha level used to determine statistical significance.
`beta_level`	Type II error probability (power/100-1)
`liberal_lambda`	Logical indicator of whether to use the liberal (cohen_f^2*(num_df+den_df)) or conservative (cohen_f^2*den_df) calculation of the noncentrality (lambda) parameter estimate. Default is FALSE.

Value

num_df = degrees of freedom for numerator, den_df = degrees of freedom for denominator, cohen_f = Cohen's f effect size, alpha_level = Type 1 error probability, beta_level = Type 2 error probability, power = Power of test (1-beta_level*100 lambda = Noncentrality parameter estimate (default = cohen_f^2*den_df, liberal = cohen_f^2*(num_df+den_df))

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Hillsdale,NJ: Lawrence Erlbaum. Aberson, C. (2019). Applied Power Analysis for the Behavioral Sciences (2nd ed.). New York,NY: Routledge.

Examples

```
design_result <- ANOVA_design(design = "2b",
n = 65,
mu = c(0,.5),
sd = 1,
plot = FALSE)
x1 = ANOVA_exact2(design_result, verbose = FALSE)
ex = power.ftest(num_df = x1$anova_table$num_df,
den_df = x1$anova_table$den_df,
cohen_f = x1$main_result$cohen_f,
alpha_level = 0.05,
liberal_lambda = FALSE)
```

---

power_oneway_ancova *Power Calculations for a one-way ANCOVA*

Description

Compute power of ANCOVA omnibus test (power_oneway_ancova) or contrast (power_oneway_ancova) for one-way (single factor), between subjects designs.

Usage

```
power_oneway_ancova(
  n = NULL,
  mu = NULL,
  n_cov = 1,
  r2 = NULL,
  sd = 1,
  alpha_level = Superpower_options("alpha_level"),
  beta_level = NULL,
  round_up = TRUE,
  type = "exact"
)
```

Arguments

n	Sample size in each condition.
mu	Vector specifying mean for each condition.
n_cov	Number of covariates.
r2	Coefficient of determination (r^2) of the combined covariates.
sd	Standard deviation for all conditions (residual SD without covariate adjustment).
alpha_level	Alpha level used to determine statistical significance.
beta_level	Type II error probability (power/100-1)
round_up	Logical indicator for whether to round up the sample size(s) to a whole number. Default is TRUE.

type	Sets the method for estimating power. "exact" will use the Shieh (2020) approach while "approx" will use the Keppel (1991) approach.

Value

dfs = degrees of freedom, N = Total sample size, n = Sample size per group/condition, n_cov = Number of covariates, mu = Mean for each condition, sd = Standard deviation, r2 = Coefficient of determination of combined covariates. alpha_level = Type 1 error probability, beta_level = Type 2 error probability, power = Power of test (1-beta_level*100 type = Method (Shieh or Keppel) for estimating power

References

Keppel, G. (1991). Design and Analysis A Researcher's Handbook. 3rd Edition. Prentice Hall. Englewood Cliffs, New Jersey. See pages 323 - 324. Shieh, G. (2017). Power and sample size calculations for contrast analysis in ANCOVA. Multivariate behavioral research, 52(1), 1-11. Shieh, G. (2020). Power analysis and sample size planning in ANCOVA designs. Psychometrika, 85(1), 101-120.

Examples

```
# Example from Table 1 Shieh 2020
power_oneway_ancova(mu = c(400, 450, 500), n = c(21,21,21),
r2 = .1^2, sd = 100)
```

---

power_oneway_between *Analytic power calculation for one-way between designs.*

Description

Analytic power calculation for one-way between designs.

Usage

```
power_oneway_between(design_result, alpha_level = 0.05)
```

Arguments

design_result	Output from the ANOVA_design function
alpha_level	Alpha level used to determine statistical significance

Value

mu = means

sigma = standard deviation

n = sample size

alpha_level = alpha level

Cohen_f = Cohen f

f_2 = Cohen's f^2

lambda = lambda

F_critical = Critical F-value

power = power

df1 = degrees of freedom for the effect

df2 = degrees of freedom of the error

eta_p_2 = partial eta-squared

mean_mat = matrix of the means

References

too be added

Examples

```
## Set up a within design with one factor with 2 levels,
## 40 participants (woh do all conditions), and standard deviation of 2
## with a mean pattern of 1, 0, 1, conditions labeled 'condition'
## with names for levels of "cheerful", "neutral", "sad"
design_result <- ANOVA_design(design = "3b", n = 40, mu = c(1, 0, 1),
      sd = 2, labelnames = c("condition", "cheerful", "neutral", "sad"))
power_result <- power_oneway_between(design_result, alpha_level = 0.05)
```

---

power_oneway_within *Analytic power calculation for one-way within designs.*

Description

Analytic power calculation for one-way within designs.

Usage

```
power_oneway_within(design_result, alpha_level = 0.05)
```

Arguments

design_result	Output from the ANOVA_design function
alpha_level	Alpha level used to determine statistical significance

Value

mu = means

sigma = standard deviation

n = sample size

alpha_level = alpha level

Cohen_f = Cohen's f

f_2 = Cohen's f squared

lambda = lambda

F_critical = Critical F-value

power = power

df1 = degrees of freedom for the effect

df2 = degrees of freedom of the error

eta_p_2 = partial eta-squared

mean_mat = matrix of the means

References

too be added

Examples

```
## Set up a within design with 3 factors,
## with correlation between observations of 0.8,
## 40 participants (who do all conditions), and standard deviation of 2
## with a mean pattern of 1, 0, 1, conditions labeled 'condition' and
## 'voice', with names for levels of "cheerful", "neutral", "sad".
design_result <- ANOVA_design(design = "3w", n = 40, r = 0.8,
      mu = c(1, 0, 1), sd = 2,
      labelnames = c("condition", "cheerful", "neutral", "sad"))
power_result <- power_oneway_within(design_result, alpha_level = 0.05)
```

---

power_standardized_alpha

Optimizing function to achieve desired power based on a standardized alpha level.

Description

Because the standardized alpha depends on the sample size (N), and the power depends on the sample size, deciding upon the sample size to achieve a desired power requires an iterative procedure. Increasing the sample size reduces the standardized alpha, which requires an increase in the sample size for the power analysis, which reduces the standardized alpha. This function takes a power analysis function that outputs the power as a function of the desired power, the alpha level, as a function of N(x).

Usage

```
power_standardized_alpha(
  power_function,
  alpha = 0.05,
  power = 0.8,
  standardize_N = 100,
  verbose = Superpower_options("verbose")
)
```

Arguments

`power_function`	Function that outputs the power, calculated with an analytic function.
`alpha`	The unstandardized alpha level (e.g., 0.05), independent of the sample size.
`power`	The desired power, i.e., the outcome of the power calculation you would like to achieve.
`standardize_N`	The sample size you want to use to standardize the alpha level for. Defaults to 100 (based on Good, 1982).
`verbose`	Set to FALSE to not print results (default = TRUE)

Value

List of 3 objects: a_stan = standardized alpha, N = sample size, and objective = for the weighted combined error rate.

References

Good, I. J. (1982). C140. Standardized tail-area probabilities. Journal of Statistical Computation and Simulation, 16(1), 65–66. <https://doi.org/10.1080/00949658208810607>

Examples

```
res <- power_standardized_alpha(power_function = "pwr::pwr.t.test(d = 0.3,
n = x, sig.level = a_stan, type = 'two.sample',
alternative = 'two.sided')$power", power = 0.9, alpha = 0.05)
res$N
```

`power_threeway_between`

Analytic power calculation for three-way between designs.

Description

Analytic power calculation for three-way between designs.

Usage

```
power_threeway_between(design_result, alpha_level = 0.05)
```

Arguments

`design_result`	Output from the ANOVA_design function
`alpha_level`	Alpha level used to determine statistical significance (default to 0.05)

Value

mu = means

sigma = standard deviation

n = sample size

alpha_level = alpha level

Cohen_f_A = Cohen's f for main effect A

Cohen_f_B = Cohen's f for main effect B

Cohen_f_C = Cohen's f for main effect C

Cohen_f_AB = Cohen's f for the A*B interaction

Cohen_f_AC = Cohen's f for the A*C interaction

Cohen_f_BC = Cohen's f for the B*C interaction

Cohen_f_ABC = Cohen's f for the A*B*C interaction

f_2_A = Cohen's f squared for main effect A

f_2_B = Cohen's f squared for main effect B

f_2_C = Cohen's f squared for main effect C

f_2_AB = Cohen's f squared for A*B interaction

f_2_AC = Cohen's f squared for A*C interaction

f_2_BC = Cohen's f squared for B*C interaction

f_2_ABC = Cohen's f squared for A*B*C interaction

lambda_A = lambda for main effect A

lambda_B = lambda for main effect B

lambda_C = lambda for main effect C

lambda_AB = lambda for A*B interaction

lambda_AC = lambda for A*C interaction

lambda_BC = lambda for B*C interaction

lambda_ABC = lambda for A*B*C interaction

critical_F_A = critical F-value for main effect A

critical_F_B = critical F-value for main effect B

critical_F_C = critical F-value for main effect C

critical_F_AB = critical F-value for A*B interaction critical_F_AC = critical F-value for A*C interaction

critical_F_BC = critical F-value for B*C interaction

critical_F_ABC = critical F-value for A*B*C interaction

power_A = power for main effect A

power_B = power for main effect B

power_C = power for main effect C

power_AB = power for A*B interaction

power_AC = power for A*C interaction

power_BC = power for B*C interaction

power_ABC = power for A*B*C interaction

df_A = degrees of freedom for main effect A

df_B = degrees of freedom for main effect B

df_C = degrees of freedom for main effect C

df_AB = degrees of freedom for A*B interaction

df_AC = degrees of freedom for A*C interaction

df_BC = degrees of freedom for B*C interaction

df_ABC = degrees of freedom for A*B*C interaction

df_error = degrees of freedom for error term

eta_p_2_A = partial eta-squared for main effect A

eta_p_2_B = partial eta-squared for main effect B

eta_p_2_C = partial eta-squared for main effect C

eta_p_2_AB = partial eta-squared for A*B interaction

eta_p_2_AC = partial eta-squared for A*C interaction

eta_p_2_BC = partial eta-squared for B*C interaction

eta_p_2_ABC = partial eta-squared for A*B*C interaction

mean_mat = matrix of the means

References

to be added

Examples

```
design_result <- ANOVA_design(design = "2b*2b*2b", n = 40,
      mu = c(1, 0, 1, 0, 0, 1, 1, 0), sd = 2,
      labelnames = c("condition", "cheerful", "sad",
      "voice", "human", "robot", "color", "green", "red"))
power_result <- power_threeway_between(design_result, alpha_level = 0.05)
```

`power_twoway_between` *Analytic power calculation for two-way between designs.*

Description

Analytic power calculation for two-way between designs.

Usage

```
power_twoway_between(design_result, alpha_level = 0.05)
```

Arguments

`design_result`	Output from the ANOVA_design function
`alpha_level`	Alpha level used to determine statistical significance

Value

mu = means

sigma = standard deviation

n = sample size

alpha_level = alpha level

Cohen_f_A = Cohen's f for main effect A

Cohen_f_B = Cohen's f for main effect B

Cohen_f_AB = Cohen's f for the A*B interaction

f_2_A = Cohen's f squared for main effect A

f_2_B = Cohen's f squared for main effect B

f_2_AB = Cohen's f squared for A*B interaction

lambda_A = lambda for main effect A

lambda_B = lambda for main effect B

lambda_AB = lambda for A*B interaction

critical_F_A = critical F-value for main effect A

critical_F_B = critical F-value for main effect B

critical_F_AB = critical F-value for A*B interaction

power_A = power for main effect A

power_B = power for main effect B

power_AB = power for A*B interaction

df_A = degrees of freedom for main effect A

df_B = degrees of freedom for main effect B

df_AB = degrees of freedom for A*B interaction

df_error = degrees of freedom for error term

eta_p_2_A = partial eta-squared for main effect A

eta_p_2_B = partial eta-squared for main effect B

eta_p_2_AB = partial eta-squared for A*B interaction

mean_mat = matrix of the means

References

too be added

Examples

```
design_result <- ANOVA_design(design = "2b*2b", n = 40, mu = c(1, 0, 1, 0),
     sd = 2, labelnames = c("condition", "cheerful", "sad",
     "voice", "human", "robot"))
power_result <- power_twoway_between(design_result, alpha_level = 0.05)
```

`p_standardized` *Compute standardized alpha level based on unstandardized alpha level and the number of observations N.*

Description

Compute standardized alpha level based on unstandardized alpha level and the number of observations N.

Usage

```
p_standardized(p, N, standardize_N = 100)
```

Arguments

`p`	The observed p-value.
`N`	The number of observations (e.g., the sample size) in the dataset
`standardize_N`	The number of observations (e.g., the sample size) you want to use to standardize the alpha level for. Defaults to 100 (base on Good, 1982).

References

Good, I. J. (1982). C140. Standardized tail-area probabilities. Journal of Statistical Computation and Simulation, 16(1), 65–66. <https://doi.org/10.1080/00949658208810607>

Examples

```
## Check it yields .05 for N = 100:
p_standardized(p = 0.05, N = 100)
## Check it yields .05 for N = 200, p = 0.03535534:
p_standardized(p = 0.03535534, N = 200)
## What is a standardized p-value for p = .05 and N = 200?
p_standardized(p = 0.05, N = 200)
## You can change the standardization N, repeating the example above:
p_standardized(p = 0.05, N = 100, standardize_N = 200)
```

---

sim_result-methods *Methods for sim_result objects*

Description

Methods defined for objects returned from the ANOVA_exact, ANOVA_exact2, and ANOVA_power functions.

Usage

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

## S3 method for class 'sim_result'
plot(x, ...)

## S3 method for class 'sim_result'
confint(object, parm = "main_results", level = 0.95, ...)
```

Arguments

`x`	object of class `sim_result` as returned from one of the simulation functions in Superpower.
`...`	further arguments passed through, see description of return value
`object`	Result returned from ANOVA_power (only applicable argument for confint)
`parm`	Argument for confint. Select what results from the simulation to return with confidence intervals. Options currently include: main_results (default), pc_results, manova_results, and emm_results.
`level`	Argument for confint. Confidence level for binomial proportion confidence intervals (Wilson, 1927). Default is .95.

Value

`print` Prints short summary of the simulation result

`plot` Returns `meansplot` or a plot of the distribution of p-values depending on whether an exact or Monte Carlo simulation was performed

`confint` Returns confidence intervals for the selected result from ANOVA_power

References

Wilson, E. (1927). Probable Inference, the Law of Succession, and Statistical Inference. Journal of the American Statistical Association, 22(158), 209-212. doi:10.2307/2276774

`Superpower_options` *Set/get global Superpower options*

Description

Global Superpower options are used, for example, by `ANOVA_exact` (et al.) and `ANOVA_power`. But can be changed in each functions directly using an argument (which has precedence over the global options).

Usage

```
Superpower_options(...)
```

Arguments

`...`	One of four: (1) nothing, then returns all options as a list; (2) a name of an option element, then returns its' value; (3) a name-value pair which sets the corresponding option to the new value (and returns nothing), (4) a list with option-value pairs which sets all the corresponding arguments. The example show all possible cases.

Details

The following arguments are currently set:

- `verbose` should verbose (printed results) be set to true? Default is `TRUE`.
- `emm` Option to perform analysis of estimated marginal means. Default is `FALSE`.
- `emm_model` Model type ("multivariate", or "univariate") for estimated marginal means. Default is `"multivariate"`.
- `contrast_type` The type of comparison for the estimated marginal means. Default is `"pairwise"`. See ?emmeans::'contrast-methods' for more details on acceptable methods.
- `plot` Option to automatically print plots. Default is `FALSE`.
- `alpha_level` Alpha level used to determine statistical significance. Default is .05.
- `correction` Option to set a correction for sphericity violations. Default is no correction. This can be set to "none", "GG" Greenhouse-Geisser, and "HF" Huynh-Feldt
- `liberal_lambda` Option to set a logical indicator of whether to use the liberal (cohen_f^2*(num_df+den_df)) or conservative (cohen_f^2*den_df) calculation of the noncentrality (lambda) parameter estimate. Default is FALSE.

Value

depends on input, see above.

Note

All options are saved in the global R options with prefix Superpower.

Index