

Package: ggpower (via r-universe)

September 23, 2026

Title Publication-Ready Power Analysis and Visualization

Version 0.1.3

Author Yaoxiang Li [aut, cre]

Maintainer Yaoxiang Li <liyaoxiang@outlook.com>

Description Provides statistical power analysis and sample size calculations for t-tests, ANOVA, regression, chi-square, proportion, correlation, nonparametric, biomarker, and clinical trial designs. Includes a scriptable API via `power_compute()`, publication-ready 'ggplot2' visualizations, and an optional 'Shiny' application.

URL <https://github.com/YaoxiangLi/ggpower>,
<https://yaoxiangli.github.io/ggpower/>

BugReports <https://github.com/YaoxiangLi/ggpower/issues>

License GPL (>= 3)

Imports ggplot2, shiny

Encoding UTF-8

LazyData true

Suggests bs4Dash, config, golem, knitr, rmarkdown, spelling, testthat
(>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

Repository <https://yaoxiangli.r-universe.dev>

Date/Publication 2026-07-25 00:29:59 UTC

RemoteUrl <https://github.com/yaoxiangli/ggpower>

RemoteRef HEAD

RemoteSha 6c2edf163e4c5398ea931bbbec3e32c0abf361a1

Contents

effect_size_d	2
effect_size_f	3
effect_size_f2	3
effect_size_f2_increase	4
effect_size_h	4
effect_size_q	5
effect_size_w	5
eta2_from_f	6
format_result_html	6
ggpower_calculator	7
ggpower_result	7
ggpower_t_one_sample	8
ggpower_tests	9
ggpower_ttest	9
odds_ratio_from_probs	10
plot_distribution	10
plot_power_curve	11
power_compute	11
power_t_one_sample	12
power_t_paired	13
power_t_two_sample	13
r2_from_f2	14
run_app	14
save_power_plot	15
theme_ggpower	16
Index	17

effect_size_d	<i>Compute Cohen's d from means and standard deviation</i>
---------------	--

Description

Compute Cohen's d from means and standard deviation

Usage

effect_size_d(mean_h1, mean_h0 = 0, sd)

Arguments

- | | |
|---------|---|
| mean_h1 | Mean assumed under H1. |
| mean_h0 | Mean assumed under H0 or comparison group mean. |
| sd | Common standard deviation. |

Value

Numeric effect size d.

effect_size_f

Compute Cohen's f from eta-squared

Description

Compute Cohen's f from eta-squared

Usage

effect_size_f(eta2)

Arguments

eta2 Proportion of variance explained.

Value

Numeric effect size f.

effect_size_f2

Compute Cohen's f2 from R-squared

Description

Compute Cohen's f2 from R-squared

Usage

effect_size_f2(r2)

Arguments

r2 Coefficient of determination.

Value

Numeric effect size f2.

`effect_size_f2_increase`*Compute f2 for an R-squared increase*

Description

Compute f2 for an R-squared increase

Usage

```
effect_size_f2_increase(r2_full, r2_reduced)
```

Arguments

`r2_full` R-squared for the full model.

`r2_reduced` R-squared for the reduced model.

Value

Numeric effect size f2.

`effect_size_h`*Compute Cohen's h for two proportions*

Description

Compute Cohen's h for two proportions

Usage

```
effect_size_h(p1, p2)
```

Arguments

`p1, p2` Proportions.

Value

Numeric h.

effect_size_q	<i>Compute Cohen's q for two correlations</i>
---------------	---

Description

Compute Cohen's q for two correlations

Usage

```
effect_size_q(r1, r2)
```

Arguments

r1, r2 Correlations to compare.

Value

Numeric Fisher-z difference q.

effect_size_w	<i>Compute chi-square effect size w</i>
---------------	---

Description

Compute chi-square effect size w

Usage

```
effect_size_w(p0, p1)
```

Arguments

p0 Probabilities under H0.
p1 Probabilities under H1.

Value

Numeric effect size w.

eta2_from_f	<i>Convert Cohen's f to eta-squared</i>
-------------	---

Description

Convert Cohen's f to eta-squared

Usage

```
eta2_from_f(f)
```

Arguments

f	Cohen's f.
---	------------

Value

Numeric eta-squared value.

format_result_html	<i>Format a ggpower result as structured HTML for Shiny UI</i>
--------------------	--

Description

Format a ggpower result as structured HTML for Shiny UI

Usage

```
format_result_html(x)
```

Arguments

x	A 'ggpower_result' object.
---	----------------------------

Value

A 'shiny.tag' list for 'renderUI'.

ggpower_calculator	Evaluate a ggpower calculator script
--------------------	--------------------------------------

Description

Evaluates distribution-function calculator expressions, including helpers such as `zcdf()`, `tinv()`, `ncfcdf()`, and `binocdf()`.

Usage

```
ggpower_calculator(script)
```

Arguments

script	Character calculator script with arithmetic, assignments, comments, and supported distribution helper functions.
--------	--

Value

The value of the final expression.

Examples

```
ggpower_calculator("x <- 2^3\nx + zinv(.975)")
```

ggpower_result	Create a ggpower result object
----------------	--------------------------------

Description

Creates the common result object used by the scriptable API and Shiny GUI.

Usage

```
ggpower_result(test, analysis, inputs, outputs, notes = character(),  
  distribution = list())
```

Arguments

test	Character label for the selected test.
analysis	Character label for the selected analysis mode.
inputs	Named list of input parameters.
outputs	Named list of computed output parameters.
notes	Character vector with method notes or assumptions.
distribution	Named list describing the H0/H1 distributions.

Value

An object of class `ggpower_result`.

`ggpower_t_one_sample` *Plot Power Curve for a One-Sample t-Test*

Description

This function creates a `ggplot2` power curve for a one-sample t test.

Usage

```
ggpower_t_one_sample(  
  d,  
  alpha = 0.05,  
  n_range = seq(20, 100, by = 5),  
  tails = "two"  
)
```

Arguments

<code>d</code>	Numeric. The effect size (d).
<code>alpha</code>	Numeric. The significance level (default 0.05).
<code>n_range</code>	Numeric vector. A vector of total sample sizes (default is <code>seq(20, 100, by = 5)</code>).
<code>tails</code>	Character. "two" or "one".

Value

A `ggplot` object showing the power curve.

Examples

```
# Plot power curve for d = 0.5 over sample sizes from 20 to 100  
ggpower_t_one_sample(d = 0.5, alpha = 0.05, n_range = seq(20, 100, by = 5))
```

ggpower_tests	List supported statistical power tests
---------------	--

Description

List supported statistical power tests

Usage

```
ggpower_tests(domain = NULL, module = NULL)
```

Arguments

domain	Optional character vector to filter by domain ('general', 'biomarker', 'pharma').
module	Optional character vector to filter by app module ('workspace', 'biomarker', 'clinical').

Value

A data frame describing tests available to 'power_compute()'.

ggpower_ttest	Plot Power Curve for a Two-Sample t-Test
---------------	--

Description

This function creates a ggplot2 power curve for a two-sample t test.

Usage

```
ggpower_ttest(d, alpha = 0.05, n_range = seq(10, 100, by = 5), tails = "two")
```

Arguments

d	Numeric. The effect size (Cohen's d).
alpha	Numeric. The significance level (default 0.05).
n_range	Numeric vector. A vector of sample sizes per group (default is seq(10, 100, by = 5)).
tails	Character. "two" or "one".

Value

A ggplot object showing the power curve.

Examples

```
# Create a power curve for d = 0.5 over a range of sample sizes per group
ggpower_ttest(d = 0.5, alpha = 0.05, n_range = seq(10, 100, by = 5))
```

odds_ratio_from_probs *Compute odds ratio from two probabilities*

Description

Compute odds ratio from two probabilities

Usage

```
odds_ratio_from_probs(p0, p1)
```

Arguments

p0 Baseline event probability.
p1 Event probability under H1.

Value

Numeric odds ratio.

plot_distribution *Plot H0 and H1 distributions*

Description

Builds a publication-ready distribution overlay for a computed power-analysis result.

Usage

```
plot_distribution(result)
```

Arguments

result A ggpower_result object.

Value

A ggplot object.

Examples

```
result <- power_compute("t_one_sample", "post_hoc", d = 0.5, n = 40)
plot_distribution(result)
```

plot_power_curve	<i>Plot a power curve</i>
------------------	---------------------------

Description

Builds a publication-ready power curve for a selected ggpower test.

Usage

```
plot_power_curve(test, n_values, analysis = "post_hoc", ...)
```

Arguments

test	Character test id.
n_values	Numeric vector of total sample sizes.
analysis	Power analysis mode used for fixed parameters.
...	Test-specific fixed parameters.

Value

A ggplot object.

Examples

```
plot_power_curve("t_one_sample", n_values = c(20, 30, 40), d = 0.5)
```

power_compute	<i>Compute statistical power analyses</i>
---------------	---

Description

Runs a power analysis using the shared ggpower compute engine. The function supports classical test families and analysis modes.

Usage

```
power_compute(test, analysis = "post_hoc", ...)
```

Arguments

test	Character test id. Use ggpower_tests() to list available ids.
analysis	One of "a_priori", "compromise", "criterion", "post_hoc", or "sensitivity".
...	Test-specific input parameters.

Value

A ggpower_result list with components test, analysis, inputs, outputs, and optional notes and distribution. The outputs element contains the solved quantities (for example sample size, power, or effect size depending on the analysis mode). See [ggpower_result](#).

Examples

```
power_compute("t_one_sample", "a_priori", d = 0.625, alpha = 0.05,
  power = 0.95, tails = "one")
```

power_t_one_sample	<i>Compute Power for a One-Sample t-Test</i>
--------------------	--

Description

Calculates the power for a one-sample t-test given the effect size (d), total sample size (n), and significance level (alpha).

Usage

```
power_t_one_sample(d, n, alpha = 0.05, tails = "two")
```

Arguments

d	Numeric. The effect size (difference from the constant divided by sigma).
n	Integer. Total sample size.
alpha	Numeric. The significance level (default is 0.05).
tails	Character. "two" for a two-tailed test or "one" for a one-tailed test.

Value

Numeric. The computed power (1 - beta).

Examples

```
# Calculate power for an effect size of 0.5 with n = 40 subjects
power_t_one_sample(d = 0.5, n = 40, alpha = 0.05)
```

power_t_paired	<i>Compute power for a paired-samples t-test</i>
----------------	--

Description

Computes achieved power for a paired-samples t-test using the noncentral t kernel.

Usage

```
power_t_paired(d, n, alpha = 0.05, tails = "two")
```

Arguments

d	Numeric paired-samples effect size dz.
n	Integer number of pairs.
alpha	Numeric significance level.
tails	Character, "two" or "one".

Value

Numeric power.

Examples

```
power_t_paired(d = 0.5, n = 40)
```

power_t_two_sample	<i>Compute Power for a Two-Sample t-Test (Equal Sample Sizes)</i>
--------------------	---

Description

This function calculates the power for a two-sample t-test when the two groups have equal sample sizes.

Usage

```
power_t_two_sample(d, n_per_group, alpha = 0.05, tails = "two", n2 = NULL)
```

Arguments

d	Numeric. The effect size (Cohen's d).
n_per_group	Integer. The sample size per group.
alpha	Numeric. The significance level (default is 0.05).
tails	Character. "two" for a two-tailed test or "one" for a one-tailed test.
n2	Optional second-group sample size. If omitted, equal group sizes are used.

Value

Numeric. The computed power (1 - beta).

Examples

```
# Compute power for an effect size d = 0.5 with 30 subjects per group
power_t_two_sample(d = 0.5, n_per_group = 30)
```

r2_from_f2	<i>Convert f2 to R-squared</i>
------------	--------------------------------

Description

Convert f2 to R-squared

Usage

```
r2_from_f2(f2)
```

Arguments

f2 Cohen's f2.

Value

Numeric R-squared value.

run_app	<i>Run the Shiny Application</i>
---------	----------------------------------

Description

Run the Shiny Application

Usage

```
run_app(
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

Arguments

onStart	A function that will be called before the app is actually run. This is only needed for shinyAppObj, since in the shinyAppDir case, a global.R file can be used for this purpose.
options	Named options that should be passed to the runApp call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify width and height parameters which provide a hint to the embedding environment about the ideal height/width for the app.
enableBookmarking	Can be one of "url", "server", or "disable". The default value, NULL, will respect the setting from any previous calls to enableBookmarking() . See enableBookmarking() for more information on bookmarking your app.
uiPattern	A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
...	arguments to pass to golem_opts. See ‘?golem::get_golem_options’ for more details.

Value

A Shiny application object (class "shiny.appobj"). Launch the GUI with 'shiny::runApp(run_app())' or from the development helper described in the package overview.

save_power_plot	<i>Save a ggpower plot</i>
-----------------	----------------------------

Description

Exports publication-ready ggpower plots through ggplot2::ggsave().

Usage

```
save_power_plot(plot, filename, width = 7, height = 5, dpi = 320)
```

```
save_distribution_plot(plot, filename, width = 7, height = 5, dpi = 320)
```

Arguments

plot	A ggplot object.
filename	Output filename.
width, height	Plot dimensions.
dpi	Resolution for raster outputs.

Value

The filename invisibly.

theme_ggpower	<i>Publication-ready ggpower theme</i>
---------------	--

Description

Provides consistent typography, spacing, and grid styling for ggpower figures.

Usage

```
theme_ggpower(base_size = 12, base_family = "")
```

Arguments

- base_size Base font size.
- base_family Base font family.

Value

A ggplot2 theme.

Index

effect_size_d, [2](#)
effect_size_f, [3](#)
effect_size_f2, [3](#)
effect_size_f2_increase, [4](#)
effect_size_h, [4](#)
effect_size_q, [5](#)
effect_size_w, [5](#)
enableBookmarking(), [15](#)
eta2_from_f, [6](#)

format_result_html, [6](#)

ggpower_calculator, [7](#)
ggpower_result, [7](#), [12](#)
ggpower_t_one_sample, [8](#)
ggpower_tests, [9](#)
ggpower_ttest, [9](#)

odds_ratio_from_probs, [10](#)

plot_distribution, [10](#)
plot_power_curve, [11](#)
power_compute, [11](#)
power_t_one_sample, [12](#)
power_t_paired, [13](#)
power_t_two_sample, [13](#)
print.ggpower_result (ggpower_result), [7](#)

r2_from_f2, [14](#)
run_app, [14](#)

save_distribution_plot
 (save_power_plot), [15](#)
save_power_plot, [15](#)

theme_ggpower, [16](#)