Package 'RChiSquare'

November 26, 2026

Type Package

Title Chi-Square Tests, Power and Sample Size for Discrete and Continuous Data in One, Two or Three Dimensions

Version 1.0.0

Description Cressie-Read power-divergence goodness-of-fit and two-sample tests for one, two and three dimensional data. Continuous data support equal-space and equal-probability partitions, simulation power, and analytic fixed-alternative power. Discrete goodness-of-fit tests support raw or value-count data with explicit support, expected-count cell merging, and analytic and simulation power calculations. The package includes a routine for sample size estimation and planning as well as a 'shiny' app for one-dimensional data. For references see the vignette.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, graphics, utils, parallel, ggplot2, shiny

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

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Wolfgang Rolke [aut, cre] (ORCID: <https://orcid.org/0000-0002-3514-726X>)

Maintainer Wolfgang Rolke <wolfgang.rolke@upr.edu>

Repository CRAN

Date/Publication 2026-11-26 11:00:14 UTC

Contents

RChiSquare-package 2

as_htest . . . 3
chi_app . . . 3
chi_cells . . . 4
chi_power . . . 4
chi_power_table . . . 8
chi_sample_size . . . 8
chi_test . . . 10
compare_power . . . 13
plot.RChiSquare_power_curve . . . 14
summary.chi_sample_size . . . 15
summary.RChiSquare_power . . . 16
summary.RChiSquare_test . . . 16

Index **17**

`RChiSquare-package` *RChiSquare: Chi-Square Tests, Power and Sample Size*

Description

Cressie–Read power-divergence (chi-square) goodness-of-fit and two-sample tests for discrete and continuous data in one, two and three dimensions, with analytic and simulation-based power and sample-size calculations.

Details

The main functions are

- `chi_test()`: tests; `summary()`, `chi_cells()` and `as_htest()` inspect the result;
- `chi_power()`: power, for one alternative or a power curve; `chi_power_table()` and `compare_power()` summarize and compare results;
- `chi_sample_size()`: sample size for a target power;
- `chi_app()`: a Shiny interface for one-dimensional tests.

See `vignette("getting-started", package = "RChiSquare")`.

Author(s)

Maintainer: Wolfgang Rolke <wolfgang.rolke@upr.edu> (ORCID)

Authors:

- Wolfgang Rolke <wolfgang.rolke@upr.edu> (ORCID)

as_htest *Convert a chi-square test result to a standard htest object*

Description

Convert a chi-square test result to a standard htest object

Usage

```
as_htest(object, partition = c("EP", "ES"))
```

Arguments

object	A result of chi_test().
partition	For continuous data, "EP" or "ES".

Value

An object of class htest, as returned by stats::chisq.test(), for use with code that expects standard test results.

Examples

```
set.seed(1)
as_htest(chi_test(rnorm(100), "pnorm"), "EP")
```

chi_app *Launch a guided Shiny interface for chi-square tests*

Description

A CSV-based interface to chi_test(). The example table changes with the problem and data type; uploaded data are previewed before testing. Custom R expressions are evaluated locally and should only be used with code the user trusts.

Usage

```
chi_app()
```

Value

Invisibly returns the Shiny application object after launching it.

chi_cells *Cell-level results of a chi-square test*

Description

Returns the final (merged) cells of a test from `chi_test()` with observed and expected counts, Pearson residuals, and each cell's contribution to the test statistic. The columns are the same for every data type and dimension.

Usage

```
chi_cells(object, partition = c("EP", "ES"))
```

Arguments

`object`	A result of `chi_test()`.
`partition`	For continuous data, `"EP"` or `"ES"`; ignored for discrete data, which have a single partition.

Value

A data frame with one row per final cell. Goodness-of-fit tests give `cell`, `observed`, `expected`, `residual` and `contribution`; two-sample tests give `cell`, `x`, `y`, `expected_x`, `expected_y`, `residual_x`, `residual_y` and `contribution`. Contributions sum to the statistic. Multidimensional cells are merged regions `R1`, `R2`, ...; their elementary grid cells are listed in `attr(object, "cells")`.

Examples

```
set.seed(1)
fit <- chi_test(rnorm(100), "pnorm")
chi_cells(fit, "EP")
```

chi_power *Automatically select a chi-square power routine*

Description

Power of the chi-square tests of `chi_test()`, by the analytic fixed-alternative approximation or by simulation, for a single alternative or a power curve over scenario parameters. Like `stats::power.t.test()`, the result prints compactly; `summary()` and `chi_power_table()` give more detail.

Usage

```
chi_power(
  rxy,
  y = NULL,
  ...,
  type = c("auto", "continuous", "discrete"),
  dimension = c("auto", "1D", "MD"),
  problem = c("auto", "gof", "2sample"),
  alpha = 0.05,
  nbins = NULL,
  bins = NULL,
  minexpcount = 5,
  pnull = NULL,
  qnull = NULL,
  phat = NULL,
  pstart = NULL,
  pdf = NULL,
  pmnull = NULL,
  pmalt = NULL,
  support = NULL,
  alt_cdf = NULL,
  alt_pdf = NULL,
  range = NULL,
  maxbins = NULL,
  family = "pearson",
  lambda = NULL,
  method = NULL,
  B = 1000,
  par1 = NULL,
  par2 = NULL,
  p_method = NULL,
  B_permutation = NULL,
  rnull = NULL,
  partition = c("both", "ES", "EP"),
  cores = NULL,
  parallel_threshold = 30,
  seed = NULL,
  verbose = getOption("RChiSquare.verbose", TRUE)
)
```

Arguments

`rxy`	Alternative-data generator; for two-sample tests return list(x=...,y=...).
`y`	For goodness-of-fit, the null CDF as a function or by name (`"pnorm"`); distribution parameters go in `....`
`...`	Parameters of a null distribution given by name, the sample sizes `n` (and `m` for two-sample problems), or further arguments for the selected routine.

type	Data type: auto, continuous, or discrete.
dimension	Dimension: auto, 1D, or MD.
problem	Problem: auto, gof, or 2sample.
alpha	Significance level.
nbins	Requested number of bins; default depends on the selected routine.
bins	Custom continuous bin boundaries; requires selected power routine support.
minexpcount	Minimum expected cell count for merging.
pnull	Null cumulative distribution function for GOF; alternative to y.
qnull	Null quantile function for continuous 1D GOF; not needed for a null given by name or a CDF from R passed as a function (see chi_test()).
phat	Function estimating null parameters from data or, for a null given by name, the names of the parameters to estimate.
pstart	Starting values for the null parameters of a composite GOF null; the final estimates minimize the divergence on the final cells. A null CDF/PMF with parameter arguments that have no default values requires phat or pstart.
pdf	Null probability mass/density function for supported discrete GOF routines.
pmnull	Null probability mass function for supported MD discrete GOF.
pmalt	Alternative PMF for multidimensional discrete GOF.
support	Discrete support specification for applicable GOF tests.
alt_cdf	Alternative CDF for analytic power where supported.
alt_pdf	Alternative PMF for discrete GOF where supported.
range	Known support: c(L, U), or a 2-by-d matrix for multidimensional data (a length-2 vector applies to every coordinate).
maxbins	Maximum number of cells for routines that support it.
family	Divergence statistic family.
lambda	Power-divergence parameter for custom family.
method	Power method: analytic, simulation, or both where supported. When omitted, the routine default (analytic) is used for family='pearson' and simulation for other families.
B	Number of simulation replications (method = "simulation"). Analytic power does not use it: without alt_cdf (or alt_pdf, pmalt) the alternative cell probabilities are estimated from reference data whose size is chosen automatically, so that each cell probability has a standard error of about 0.001; data-dependent partitions are averaged until the Monte Carlo error of the average is below 0.002.
par1, par2	Scenario parameters, paired elementwise. They are passed in order to the alternative generator; a vector of values gives a power curve. xparam/yparam are accepted as former names.
p_method	Calibration of the simulated tests: "chisq" (asymptotic), "permutation" (two-sample tests) or "bootstrap" (goodness-of-fit tests, see chi_test()). Bootstrap power needs 2B tests rather than B^2: the null distribution of the statistic is

simulated once, from the null member closest to the alternative (the minimum-divergence fit to a large alternative sample, separately for ES and EP), and each simulated test under the alternative rejects when its bootstrap p-value against these null statistics is at most `alpha`. The result reports the estimated critical value in `critical_value`.

`B_permutation` Number of permutations per simulated test when `p_method = "permutation"`.

`rnull` Generator of null samples for bootstrap power, as in `chi_test()`; not needed for a null given by name, a PMF with support, or one-dimensional data.

`partition` For continuous data, report `"both"` partitions, or only `"ES"` or `"EP"`.

`cores` Maximum number of worker processes.

`parallel_threshold` Estimated runtime in seconds above which the calculation runs in parallel.

`seed` Random seed for reproducibility.

`verbose` Whether to report ambiguous automatic classification.

Value

A data frame of class `RChiSquare_power` (one row per method, partition and bin count), or for a vector of scenario parameters an `RChiSquare_power_curve`. Former argument names (`Range`, `Ranges`, `maxProcessor`, `parallel.threshold`, `p.method`, `B.permutation`, `xparam`, `yparam`) are still accepted.

Examples

```
# Analytic power (the default for Pearson's statistic): fitted normal
# null against a t distribution with 4 degrees of freedom.
chi_power(function(n) rt(n, 4), "pnorm", phat=c("mean","sd"),
  n=200, nbins=8, cores=1, seed=201)

# With a known alternative CDF no reference samples are needed.
chi_power(function(n) rnorm(n, 0.3), "pnorm",
  alt_cdf=function(z) pnorm(z, 0.3), n=100, nbins=c(5, 8))

# Discrete GOF: Poisson null with estimated rate, overdispersed data.
chi_power(function(n) rnbinom(n, size=5, mu=2.5), "ppois",
  phat="lambda", n=150, type="discrete", cores=1, seed=203)

# Simulated power. B=50 only keeps the example fast.
chi_power(function(n, m) list(x=rnorm(n), y=rnorm(m, 0.3)),
  n=40, m=45, nbins=4, method="simulation", B=50, cores=1, seed=202)

# Power curve over a location shift, analytic, EP partition only.
chi_power(function(n, shift) rnorm(n, shift), "pnorm", par1=c(0, .2, .4),
  n=100, nbins=6, partition="EP", cores=1, seed=204)
```

chi_power_table *Compact power estimates*

Description

Convert a chi_power result, a parameter power curve, or a named list of chi_power results to a compact table. Diagnostic information remains in the original results.

Usage

```
chi_power_table(object, alternative = NULL)
```

Arguments

object	A result of chi_power(), an RChiSquare power curve, or a list of results (one per alternative).
alternative	Optional labels for a list of results.

Value

A data frame containing power estimates.

Examples

```
pw <- chi_power(function(n) rnorm(n, 0.3), "pnorm", n=100, nbins=c(5, 8),
  cores=1, seed=2)
chi_power_table(pw)
```

chi_sample_size *Find a sample size attaining a target chi-square test power*

Description

Searches on a two-significant-digit sample-size grid. Analytic power supplies the initial estimate; method="simulation" then searches outward from that estimate using simulated power (from n_min when no analytic estimate is available, e.g. for non-Pearson families). Missing alternative CDFs are estimated from the alternative generator by the analytic GOF power routines. Simulation noise can violate monotonicity; increase B to verify a design.

Usage

```
chi_sample_size(
  rxy,
  y = NULL,
  power = 0.8,
  n_min = 20L,
  n_max = 100000L,
  ratio = 1,
  method = c("simulation", "analytic"),
  B = 1000L,
  partition = NULL,
  seed = NULL,
  verbose = FALSE,
  ...
)
```

Arguments

rxy	Alternative generator. It must take the sample size as an argument named n (for two-sample tests, n and m, returning list(x=..., y=...)), or as its first argument(s) without default values.
y	Null CDF for GOF (a function or a name such as "pnorm"), or NULL for two-sample power.
power	Desired power, strictly between zero and one (formerly target_power, which is still accepted).
n_min, n_max	Lower and upper bounds on the first sample size.
ratio	Two-sample allocation ratio m/n (ignored for GOF).
method	"analytic" returns the analytic estimate; "simulation" refines the analytic estimate using simulation.
B	Number of simulation replications; ignored by analytic power.
partition	For continuous-data power, select "ES" or "EP". Defaults to "EP" when continuous power returns both ES and EP. Specify "ES" to use equal-size bins instead. (Formerly bin_type, which is still accepted.)
seed	Seed forwarded to power routines and held constant across candidate sizes (common random numbers keep the search monotone). If NULL (default), one seed is drawn from the current RNG stream, so the search is reproducible after set.seed().
verbose	Show evaluation progress.
...	Other arguments passed to chi_power(), such as pnull, nbins, alpha, type, dimension, and problem. Supply a single bin configuration when determining sample size.

Value

An object of class chi_sample_size: a list with n, m, power (attained), target_power, reached, evaluations (a data frame), and the requested design settings.

Examples

```
# Analytic search; a known alternative CDF makes it fast.
chi_sample_size(function(n) rnorm(n, mean=0.5), "pnorm",
  alt_cdf=function(z) pnorm(z, mean=0.5), power=0.8,
  method="analytic", nbins=5, partition="EP", seed=206)

# Simulation refines the analytic estimate. B=5 only keeps the example
# fast; use B=1000 or more in practice.
chi_sample_size(function(n) rnorm(n, mean=0.5), "pnorm",
  alt_cdf=function(z) pnorm(z, mean=0.5), power=0.8,
  method="simulation", B=5, partition="EP", nbins=5,
  cores=1, seed=206)
```

---

chi_test *Chi-square goodness-of-fit and two-sample tests*

Description

One interface to Cressie–Read power-divergence tests for discrete and continuous data in one, two or three dimensions. The test is selected from the data; `problem`, `type` and `dimension` override the automatic choice. Options not applicable to the selected test produce a clear error.

Goodness-of-fit nulls can be given as a CDF function, by name (`chi_test(x, "pnorm", mean=0, sd=2)`, as in `stats::ks.test()`), or for categorical data as a vector of category probabilities `p` (as in `stats::chisq.test()`). Two samples can be given as `x` and `y` or with a formula `response ~ group`.

Usage

```
chi_test(
  x,
  y = NULL,
  ...,
  data = NULL,
  type = c("auto", "continuous", "discrete"),
  dimension = c("auto", "1D", "MD"),
  problem = c("auto", "gof", "2sample"),
  alpha = 0.05,
  nbins = NULL,
  bins = NULL,
  minexpcount = 5,
  pnull = NULL,
  qnull = NULL,
  phat = NULL,
  pstart = NULL,
  pdf = NULL,
  pmnull = NULL,
  p = NULL,
```

```
  support = NULL,
  values = NULL,
  rnull = NULL,
  range = NULL,
  maxbins = NULL,
  family = "pearson",
  lambda = NULL,
  p_method = NULL,
  B = NULL,
  partition = c("both", "ES", "EP"),
  seed = NULL,
  cores = NULL,
  verbose = getOption("RChiSquare.verbose", TRUE)
)
```

Arguments

x	Observed data: a numeric vector, matrix or data frame (rows are observations), a factor or character vector (with p), a one-, two- or three-way frequency table() with numeric category labels, another supported frequency format, or a formula response ~ group (with data) for a two-sample test. For a formula, x is the first level of group and y the second; use cbind(a, b, c) ~ group for multidimensional responses.
y	Second sample for two-sample tests; for goodness-of-fit, the null CDF as a function or by name ("pnorm" or "norm").
...	Parameters of a null distribution given by name (e.g. mean=0, sd=2), or further arguments for the selected routine.
data	Data frame for the formula interface.
type	Data type: auto, continuous, or discrete.
dimension	Dimension: auto, 1D, or MD.
problem	Problem: auto, gof, or 2sample.
alpha	Significance level for the reject indicator.
nbins	Requested number of bins; default depends on the selected routine.
bins	Custom continuous boundaries: numeric vector (1D) or list of vectors (MD).
minexpcount	Minimum expected cell count for merging.
pnull	Null CDF for goodness-of-fit (function or name); alternative to y.
qnull	Null quantile function for continuous 1D goodness-of-fit, used to build the equal-probability cells. It is not needed for a null given by name ("pnorm" uses qnorm) or for a CDF from R passed as a function (y = pnorm); otherwise the CDF is inverted numerically.
phat	Function estimating the null parameters from the data, or, for a null given by name, the names of the parameters to estimate (phat=c("mean","sd")).
pstart	Starting values for the null parameters of a composite null; the final estimates minimize the divergence on the final cells. For a null given by name, a named vector also marks those parameters as estimated. A null CDF/PMF with parameter arguments that have no default values requires phat or pstart.

pdf	Null probability mass function for discrete goodness-of-fit.
pmnull	Null probability mass function (alias of pdf for multidimensional data).
p	Null category probabilities for categorical goodness-of-fit, as in stats::chisq.test(): named by category, or in the order of the sorted distinct values of x. x may be a vector of categories or a table.
support	Discrete support specification for applicable goodness-of-fit tests.
values	Explicit discrete categories for applicable 1D tests.
rnull	Generator of null samples for p_method = "bootstrap": function(n) for a fully specified null, function(n, theta) for a composite null, returning data in the same form as x. It is not needed for a null given by name ("pnorm" uses rnorm), a PMF with support or p, or one-dimensional continuous data with qnull or pnull (inverse-transform sampling), or one-dimensional integer data with a CDF.
range	Known support: c(L, U) (either may be infinite), or a 2-by-d matrix of lower and upper limits for multidimensional data. A length-2 vector applies the same limits to every coordinate.
maxbins	Maximum number of final cells for routines that support it.
family	Divergence statistic family.
lambda	Power-divergence parameter for custom family. For a null given by name with a lambda parameter ("ppois"), lambda is that distribution parameter unless family = "custom".
p_method	How p-values are computed: "chisq" (default, the asymptotic chi-square distribution), "permutation" (two-sample tests: random relabellings of the pooled data), or "bootstrap" (goodness-of-fit tests: parametric bootstrap). The bootstrap draws samples of the observed size from the fitted null distribution and repeats the complete test on each, including parameter estimation, cell construction and merging, so it does not rely on the chi-square approximation or on its degrees of freedom. For a fully specified null it is a Monte Carlo test. With continuous data the ES and EP tests are each calibrated under their own fitted null. The p-value is (1 + #{T* >= T}) / (1 + B); the asymptotic p-values are kept in attr(result, "asymptotic_p").
B	Number of permutations ("permutation") or bootstrap samples ("bootstrap", default 999).
partition	For continuous data, report "both" partitions, or only "ES" or "EP". Choose the partition in advance: selecting the one with the smaller p-value after seeing both inflates the Type I error.
seed	Optional nonnegative integer seed for the permutations or bootstrap samples; the caller's random-number state is restored afterwards. With NULL (default) the current random-number stream is used, so set.seed() beforehand also makes the result reproducible. Asymptotic tests involve no randomness and ignore it.
cores	Maximum number of worker processes for the bootstrap.
verbose	Whether to report messages, such as removed missing values or an ambiguous automatic classification.

Value

An object of class `RChiSquare_test`: a matrix with rows `ES` and `EP` (continuous data) or a named vector (discrete data) with columns `statistic`, `p.value`, `df`, `K` (final cells) and `reject`. Printing gives a compact report; `summary()` adds fitted parameters, `chi_cells()` the cell-level counts and contributions, and `as_htest()` a standard `htest` object. Missing values are removed with a message. Former argument names (`Range`, `Ranges`, `p.method`) are still accepted.

Examples

```
set.seed(101)
x <- rnorm(80, mean=0.5, sd=1.2)
# Fully specified normal null, by name.
chi_test(x, "pnorm", mean=0.5, sd=1.2)
# Composite null: estimate mean and sd.
chi_test(x, "pnorm", phat=c("mean","sd"), nbins=6)

# Categorical goodness-of-fit with category probabilities.
colour <- sample(c("red","green","blue"), 200, replace=TRUE, prob=c(.5,.3,.2))
chi_test(colour, p=c(red=.5, green=.3, blue=.2))

# Two-sample tests, with and without a formula.
chi_test(rnorm(50), rnorm(55, mean=0.3), nbins=4)
chi_test(cbind(long, lat, depth) ~ mag >= median(mag), data=datasets::quakes,
         nbins=c(4, 4, 4), family="likelihood")

# Real data: are yearly counts of great discoveries Poisson?
fit <- chi_test(as.numeric(datasets::discoveries), "ppois", phat="lambda")
fit
summary(fit)       # adds the fitted rate and the cells
chi_cells(fit)     # cell counts, residuals and contributions
# Parametric bootstrap p-value: no chi-square approximation needed.
chi_test(as.numeric(datasets::discoveries), "ppois", phat="lambda",
         p_method="bootstrap", B=99, seed=1)

# Likelihood-ratio statistic and permutation p-values (two samples).
chi_test(rnorm(60), rnorm(60, mean=0.5), nbins=4, family="likelihood",
         p_method="permutation", B=199, seed=1)

# Two-dimensional goodness-of-fit: the joint null CDF takes a vector.
z <- cbind(rnorm(200), rnorm(200))
chi_test(z, function(u) pnorm(u[1])*pnorm(u[2]), nbins=c(3, 3))
```

`compare_power` *Compare analytic and simulated power for any RChiSquare power routine*

Description

Calls a power function once for each method, retaining its native results. A single plotting interface covers continuous/discrete GOF and two-sample tests in one or multiple dimensions. Base R graphics; no extra dependencies.

Usage

```
compare_power(
  fun,
  ...,
  plot = TRUE,
  x = NULL,
  verbose = getOption("RChiSquare.verbose", TRUE)
)
```

Arguments

fun	A power function (or its name as a string).
...	Arguments passed to fun, excluding method.
plot	Whether to draw the comparison immediately.
x	Optional numeric x coordinates for alternatives or bin counts.
verbose	Should minor results be shown?

Value

Invisibly, a list with analytic, simulation, and plot_data.

Examples

```
# Analytic and simulated power for a normal location shift.
cmp <- compare_power(chi_power, rxy=function(n) rnorm(n, mean=0.4),
  y="pnorm", alt_cdf=function(z) pnorm(z, mean=0.4),
  n=40, nbins=4, B=20, cores=1, seed=205, plot=FALSE, verbose=FALSE)
cmp$plot_data
```

---

plot.RChiSquare_power_curve

Plot a chi-square power curve

Description

Draws the power curves of a chi_power() result over its scenario parameters (par1/par2), one line per partition and bin setting, using ggplot2.

Usage

```
## S3 method for class 'RChiSquare_power_curve'
plot(x, along = c("par1", "par2", "index"), target = NULL, facet = TRUE, ...)
```

Arguments

x	A power curve returned by chi_power() with a vector of scenario parameters.
along	Horizontal axis: "par1", "par2" or "index".
target	Optional reference power drawn as a horizontal line.
facet	If TRUE, facet by the other parameter when it varies.
...	Unused.

Value

A ggplot object, which can be modified with further layers.

Examples

```
if (requireNamespace("ggplot2", quietly=TRUE)) {
  pw <- chi_power(function(n, mu) rnorm(n, mu), "pnorm",
                  par1=c(0, 0.1, 0.2, 0.3, 0.4), n=150, nbins=7,
                  cores=1, seed=1)
  plot(pw, target=0.8)
}
```

---

summary.chi_sample_size

Detailed summary of a chi-square sample-size search

Description

Detailed summary of a chi-square sample-size search

Usage

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

Arguments

object	Result from chi_sample_size().
...	Unused.

Value

Invisibly returns object.

summary.RChiSquare_power

Formatted chi-square power report

Description

Formatted chi-square power report

Usage

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

Arguments

object	A result from chi_power().
...	Unused.

Value

Called for its side effect of printing a formatted table of the power estimates (one row per method, partition and number of bins). The object of class RChiSquare_power is returned invisibly and unchanged.

summary.RChiSquare_test

Formatted chi-square test report

Description

Formatted chi-square test report

Usage

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

Arguments

object	A result from chi_test().
...	Unused.

Value

Called for its side effect of printing a formatted report: the test statistics, degrees of freedom, p-values and numbers of cells for each partition, any estimated parameters, and the cell tables. The object of class RChiSquare_test is returned invisibly and unchanged.

Index

as_htest, 3
as_htest(), *2*, *13*

chi_app, 3
chi_app(), *2*
chi_cells, 4
chi_cells(), *2*, *13*
chi_power, 4
chi_power(), *2*, *9*, *14*, *15*
chi_power_table, 8
chi_power_table(), *2*, *4*
chi_sample_size, 8
chi_sample_size(), *2*
chi_test, 10
chi_test(), *2–4*, *6*, *7*
compare_power, 13
compare_power(), *2*

plot.RChiSquare_power_curve, 14

RChiSquare *(*RChiSquare-package*)*, 2
RChiSquare-package, 2

stats::chisq.test(), *3*, *10*, *12*
stats::ks.test(), *10*
stats::power.t.test(), *4*
summary(), *2*, *4*, *13*
summary.chi_sample_size, 15
summary.RChiSquare_power, 16
summary.RChiSquare_test, 16

table(), *11*