Package 'statBasics'

August 18, 2022

Title Basic Functions to Statistical Methods Course

Version 0.2.0

Maintainer Gilberto Sassi <sassi.pereira.gilberto@gmail.com>

Description Basic statistical methods with some modifications for the course Statistical Methods at Federal University of Bahia (Brazil). All methods in this packages are explained in the text book of Montgomery and Runger (2010) <ISBN: 978-1-119-74635-5>.

Imports tibble, stats, stringr

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0), EnvStats, BSDA, purrr

Config/testthat/edition 3

NeedsCompilation no

Author Gilberto Sassi [aut, cre],
Carolina Costa Mota Paraíba [aut]

Repository CRAN

Date/Publication 2022-08-17 22:20:06 UTC

R topics documented:

`ci_1pop_bern` *Confidence interval for a population proportion*

Description

`ci_1pop_bern` can be used for obtaining the confidence intervalo for a proportion for a group.

Usage

```
ci_1pop_bern(x, n = NULL, conf_level = 0.95, type = "two.sided", na.rm = F)
```

Arguments

`x`	a vector of counts of sucesses.
`n`	a vector of counts of trials.
`conf_level`	confidence level of the returned confidence interval. Must be a single number between 0 and 1.
`type`	a character string specifying the type of confidence interval. Must be one of "two.sided" (default), "right" or "left".
`na.rm`	a logical value indicating whether `NA` values should be removed before the computation proceeds.

Details

`type` specifies the type of confidence interval. If `type` is "two.sided", the returned confidence interval is `(lower_ci, upper_ci)`. If `type` is "left", the returned confidence interval is `(lower_ci, 1)`. And, finally, if `type` is "right", the returned confidence interval is `(0, upper_ci))`.

Value

A 1 x 3 tibble with 'lower_ci', 'upper_ci', and 'conf_level' columns. Values correspond to the lower and upper bounds of the confidence interval, and to the confidence level, respectively.

Examples

```
heads <- rbinom(1, size = 100, prob = .5)
ci_1pop_bern(heads)
```

ci_1pop_exp *Confidence interval for a population mean (exponential distribution)*

Description

Confidence interval for a population mean (exponential distribution)

Usage

```
ci_1pop_exp(x, conf_level = 0.95, type = "two.sided", na.rm = F)
```

Arguments

x	a (non-empty) numeric vector.
conf_level	confidence level of the returned confidence interval. Must be a single number between 0 and 1.
type	a character string specifying the type of confidence interval. Must be one of "two.sided" (default), "right" or "left".
na.rm	a logical value indicating whether NA values should be removed before the computation proceeds.

Details

"lower_ci" and "upper_ci" are computed using pivotal quantity, as explained by Montgomery and Runger «ISBN: 978-1-119-74635-5».

Value

A 1 x 3 tibble with 'lower_ci', 'upper_ci', and 'conf_level' columns. Values correspond to the lower and upper bounds of the confidence interval, and the confidence level, respectively.

Examples

```
x <- rexp(1000)
ci_1pop_exp(x)
```

`ci_1pop_general` *Confidence interval for a population mean (general case)*

Description

Confidence interval for a population mean (general case)

Usage

```
ci_1pop_general(x, conf_level = 0.95, type = "two.sided", na.rm = F)
```

Arguments

`x`	a (non-empty) numeric vector.
`conf_level`	confidence level of the returned confidence interval. Must be a single number between 0 and 1.
`type`	a character string specifying the type of confidence interval. Must be one of "two.sided" (default), "right" or "left".
`na.rm`	a logical value indicating whether `NA` values should be removed before the computation proceeds.

Details

"lower_ci" and "upper_ci" are computed using the `t.test` function.

Value

A 1 x 3 tibble with 'lower_ci', 'upper_ci', and 'conf_level' columns. Values correspond to the lower and upper bounds of the confidence interval, and the confidence level, respectively.

Examples

```
x <- rpois(1000, lambda = 10)
ci_1pop_general(x)
```

`ci_1pop_norm` *Confidence interval for the normal distribution parameters*

Description

Confidence interval for the normal distribution parameters

Usage

```
ci_1pop_norm(
  x,
  sd_pop = NULL,
  parameter = "mean",
  conf_level = 0.95,
  type = "two.sided",
  na.rm = F
)
```

Arguments

`x`	a (non-empty) numeric vector.
`sd_pop`	a number specifying the known standard deviation of the population.
`parameter`	a character string specifying the parameter in the normal distribution. Must be one of "mean" or "variance".
`conf_level`	confidence level of the returned confidence interval. Must be a single number between 0 and 1.
`type`	a character string specifying the type of confidence interval. Must be one of "two.sided" (default), "right" or "left".
`na.rm`	a logical value indicating whether `NA` values should be removed before the computation proceeds.

Details

`type` specifies the type of confidence interval. If `type` is "two.sided", the returned confidence interval is `(lower_ci, upper_ci)` when `parameter` is "mean" or "variance". If `type` is "left", the returned confidence interval is `(lower_ci, Inf)` when `parameter` is "mean" or "variance". And, finally, is `type` is "right", the returned confidence interval is `(-Inf, upper_ci))` when `parameter` is "mean", and the returned confidence interval is `(0, upper_ci)` when `parameter` is "variance".

Value

A 1 x 3 tibble with 'lower_ci', 'upper_ci', and 'conf_level' columns. Values correspond to the lower and upper bounds of the confidence interval, and the confidence level, respectivel.

Examples

```
x <- rnorm(1000)
ci_1pop_norm(x) # confidence interval for the mean, unknown variance

x <- rnorm(1000, sd = 2)
ci_1pop_norm(x, sd_pop = 2) # confidence interval for mean, known variance

x <- rnorm(1000, sd = 5)
ci_1pop_norm(x, parameter = "variance") # confidence interval for the variance
```

---

ci_2pop_bern *Confidence interval for the difference in two population proportions*

Description

Computes the interval for different in two proportions from two distinct and independent population.

Usage

```
ci_2pop_bern(
  x,
  y,
  n_x = NULL,
  n_y = NULL,
  conf_level = 0.95,
  type = "two.sided",
  na.rm = F
)
```

Arguments

x	a (non-empty) numeric vector of 0 and 1 or a non-negative number representing number of successes.
y	a (non-empty) numeric vector of 0 and 1 or a non-negative number representing number of successes.
n_x	non-negative number of cases.
n_y	non-negative number of cases.
conf_level	confidence level of the returned confidence interval. Must be a single number between 0 and 1.
type	a character string specifying the type of confidence interval. Must be one of "two.sided" (default), "right" or "left".
na.rm	a logical value indicating whether NA values should be removed before the computation proceeds.

Details

type specifies the type of confidence interval. If type is "two.sided", the returned confidence interval is (lower_ci, upper_ci). If type is "left", the returned confidence interval is (lower_ci, Inf). And, finally, is type is "right", the returned confidence interval is (-Inf, upper_ci)).

If is.null(n_x) == T and is.null(n_y) == T, then x and y must be a numeric value of 0 and 1 and the proportions are computed using x and y. If is.null(n_x) == F and is.null(n_y) == F, then x, y, n_x and n_y must be non-negative integer scalar and x <= n_x and y <= n_y.

Value

A 1 x 3 tibble with 'lower_ci', 'upper_ci', and 'conf_level' columns. Values correspond to the lower and upper bounds of the confidence interval, and to the confidence level, respectively.

Examples

```
x <- 3
n_x <- 100
y <- 50
n_y <- 333
ci_2pop_bern(x, y, n_x, n_y)

x <- rbinom(100, 1, 0.75)
y <- rbinom(500, 1, 0.75)
ci_2pop_bern(x, y)
```

---

ci_2pop_norm *Confidence Interval for the normal distribution parameters - 2 populations*

Description

Computes the confidence interval for the difference in two population means or computes the confidence interval for the ratio of two population variances according to the parameter argument.

Usage

```
ci_2pop_norm(
  x,
  y,
  sd_pop_1 = NULL,
  sd_pop_2 = NULL,
  var_equal = FALSE,
  parameter = "mean",
  conf_level = 0.95,
  type = "two.sided",
  na.rm = F
)
```

Arguments

x	a (non-empty) numeric vector.
y	a (non-empty) numeric vector.
sd_pop_1	a number specifying the known standard deviation of the first population. Default value is NULL.
sd_pop_2	a number specifying the known standard deviation of the second population. Default value is NULL.
var_equal	a logical variable indicating whether to treat the two variances as being equal. If TRUE then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
parameter	a character string specifying the parameter in the normal distribution. Must be one of "mean" (confidence interval for mean difference) or "variance" (confidence interval for variance ratio). Default value is "mean".
conf_level	confidence level of the returned confidence interval. Must be a single number between 0 and 1.
type	a character string specifying the type of confidence interval. Must be one of "two.sided" (default), "right" or "left".
na.rm	a logical value indicating whether NA values should be removed before the computation proceeds.

Details

type specifies the type of confidence interval. If type is "two.sided", the returned confidence interval is (lower_ci, upper_ci) when parameter is "mean" or "variance". If type is "left", the returned confidence interval is (lower_ci, Inf) when parameter is "mean" or "variance". And, finally, is type is "right", the returned confidence interval is (-Inf, upper_ci)) when parameter is "mean", and the returned confidence interval is (0, upper_ci) when parameter is "variance".

Value

A 1 x 3 tibble with 'lower_ci', 'upper_ci', and 'conf_level' columns. Values correspond to the lower and upper bounds of the confidence interval, and to the confidence level, respectively.

Examples

```
x <- rnorm(1000, mean = 0, sd = 2)
y <- rnorm(1000, mean = 0, sd = 1)
# confidence interval for difference in two means, unknown variances
ci_2pop_norm(x, y)

x <- rnorm(1000, mean = 0, sd = 2)
y <- rnorm(1000, mean = 0, sd = 3)
# confidence interval for difference in two means, known variances
ci_2pop_norm(x, y, sd_pop_1 = 2, sd_pop_2 = 3)

x <- rnorm(1000, mean = 0, sd = 2)
y <- rnorm(1000, mean = 0, sd = 3)
```

```
# confidence interval for the ratoi of two population variance
ci_2pop_norm(x, y, parameter = "variance")
```

`ht_1pop_mean` *Hypothesis testing for the mean (normal distribution)*

Description

Hypothesis testing for the mean (normal distribution)

Usage

```
ht_1pop_mean(
  x,
  mu = 0,
  sd_pop = NULL,
  alternative = "two.sided",
  conf_level = NULL,
  sig_level = 0.05,
  na.rm = TRUE
)
```

Arguments

`x`	a (non-empty) numeric vector.
`mu`	a number indicating the true value of the mean. Default value is 0.
`sd_pop`	a number specifying the known standard deviation of the population. If `sd_pop == NULL`, we use the t-ttest. If `!is.null(sd_pop)`, we use the z-test. Default value is `NULL`.
`alternative`	a character string specifying the alternative hypothesis, must be one of ‘"two.sided"’ (default), ‘"greater"’ or ‘"less"’. You can specify just the initial letter.
`conf_level`	a number indicating the confidence level to compute the confidence interval. If `conf_level = NULL`, then the confidence interval is not included in the output. Default value is `NULL`.
`sig_level`	a number indicating the significance level to use in the General Procedure for Hypothesis Testing.
`na.rm`	a logical value indicating whether `NA` values should be removed before the computation proceeds.

Details

We have wrapped the `t.test` and the `BSDA::z.test` in a function as explained in the book of Montgomery and Runger (2010) <ISBN: 978-1-119-74635-5>.

Value

a `tibble` with the following columns:

statistic the value of the test statistic.

p_value the p-value of the test.

critical_value critical value in the General Procedure for Hypothesis Testing.

critical_region critical region in the General Procedure for Hypothesis Testing.

mu a number indicating the true value of the mean.

alternative character string giving the direction of the alternative hypothesis.

lower_ci lower bound of the confidence interval. It is presented only if `!is.null(con_level)`.

upper_ci upper bound of the confidence interval. It is presented only if `!is.null(con_level)`.

Examples

```
sample <- rnorm(1000, mean = 10, sd = 2) #t-test
ht_1pop_mean(sample, mu = -1) # H0: mu == -1

sample <- rnorm(1000, mean = 5, sd = 3) # z-test
ht_1pop_mean(sample, mu = 0, sd_pop = 3, alternative = 'less') # H0: mu >= 0
```

---

ht_1pop_prop *Hypothesis testing for the population proportion*

Description

One-sample test for proportion.

Usage

```
ht_1pop_prop(
  x,
  n = NULL,
  proportion = 0.5,
  alternative = "two.sided",
  conf_level = NULL,
  sig_level = 0.05,
  na.rm = TRUE
)
```

Arguments

x	a (non-empty) numeric vector indicating the number of successes. It can also be a vector with the number of successes, or it can be vector of 0 and 1.
n	a (non-empty) numeric vector indicating the number of trials. It can also be a vector with the number of trials (if x is a vector of successes), or it can be NULL (if x is a vector of 0 e 1).

`proportion`	a number between 0 e 1 indicating the value in the null hypothesis. Default value is 0.5.
`alternative`	a character string specifying the alternative hypothesis, must be one of ‘"two.sided"’ (default), ‘"greater"’ or ‘"less"’. You can specify just the initial letter.
`conf_level`	a number indicating the confidence level to compute the confidence interval. If `conf_level = NULL`, then the confidence interval is not included in the output. Default value is `NULL`.
`sig_level`	a number indicating the significance level to use in the General Procedure for Hypotheiss Testing.
`na.rm`	a logical value indicating whether `NA` values should be removed before the computation proceeds.

Value

a `tibble` with the following columns:

statistic the value of the test statistic.

p_value the p-value for the test.

critical_value critical value in the General Procedure for Hypothesis Testing.

critical_region critical region in the General Procedure for Hypothesis Testing.

proportion a number indicating the true value of the proportion.

alternative character string giving the direction of the alternative hypothesis.

lower_ci lower bound of the confidence interval. It is presented only if `!is.null(con_level)`.

upper_ci upper bound of the confidence interval. It is presented only if `!is.null(con_level)`.

Examples

```
sample <- rbinom(1, size = 100, prob = 0.75)
ht_1pop_prop(sample, proportion = 0.75, 100, conf_level = 0.99)

sample <- c(rbinom(1, size = 10, prob = 0.75),
rbinom(1, size = 20, prob = 0.75),
rbinom(1, size = 30, prob = 0.75))
ht_1pop_prop(sample, c(10, 20, 30), proportion = 0.99, conf_level = 0.90, alternative = 'less')

sample <- rbinom(100, 1, prob = 0.75)
ht_1pop_prop(sample, proportion = 0.01, conf_level = 0.95, alternative = 'greater')
```

ht_1pop_var *Hypothesis testing for the population variance*

Description

One-Sample chi-squared test on variance.

Usage

```
ht_1pop_var(
  x,
  sigma = 1,
  alternative = "two.sided",
  conf_level = NULL,
  sig_level = 0.05,
  na.rm = TRUE
)
```

Arguments

x	a (non-empty) numeric vector.
sigma	a number indicating the true value of the standard deviation in the null hypothesis. Default value is 1.
alternative	a character string specifying the alternative hypothesis, must be one of '"two.sided"' (default), '"greater"' or '"less"'. You can specify just the initial letter.
conf_level	a number indicating the confidence level to compute the confidence interval. If conf_level = NULL, then the confidence interval is not included in the output. Default value is NULL.
sig_level	a number indicating the significance level to use in the General Procedure for Hypotheiss Testing.
na.rm	a logical value indicating whether NA values should be remove before the computation proceeds.

Details

We have wrapped the EnvStats::varTest in a function as explained in the book of Montgomery and Runger (2010) <ISBN: 978-1-119-74635-5>.

Value

a tibble with the following columns:

statistic the value of the test statistic.

p_value the p-value for the test.

critical_value critical value in the General Procedure for Hypothesis Testing.

critical_region critical region in the General Procedure for Hypothesis Testing.

sigma a number indicating the true value of sigma.

alternative character string giving the direction of the alternative hypothesis.

lower_ci lower bound of the confidence interval. It is presented only if `!is.null(con_level)`.

upper_ci upper bound of the confidence interval. It is presented only if `!is.null(con_level)`.

Examples

```
sample <- rnorm(1000, mean = 10, sd = 2)
ht_1pop_var(sample, sigma = 1) # H0: sigma = 1
```

`ht_2pop_mean` *Hypothesis testing mean for two populations*

Description

Performs a hypothesis testing for the difference in means of two populations.

Usage

```
ht_2pop_mean(
  x,
  y,
  delta = 0,
  sd_pop_1 = NULL,
  sd_pop_2 = NULL,
  var_equal = FALSE,
  alternative = "two.sided",
  conf_level = NULL,
  sig_level = 0.05,
  na_rm = TRUE
)
```

Arguments

`x`	a (non-empty) numeric vector.
`y`	a (non-empty) numeric vector.
`delta`	a scalar value indicating the difference in means (Δ_0). Default value is 0.
`sd_pop_1`	a number specifying the known standard deviation of the first population. Default value is `NULL`.
`sd_pop_2`	a number specifying the known standard deviation of the second population. Default value is `NULL`.
`var_equal`	a logical variable indicating whether to treat the two variances as being equal. If `TRUE` then the pooled variance is used to estimate the variance, otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used. Default value is `FALSE`.

`alternative`	a character string specifying the alternative hypothesis, must be one of '"two.sided"' (default), '"greater"' or '"less"'.
`conf_level`	a number indicating the confidence level to compute the confidence interval. If `conf_level = NULL`, then confidence interval is not included in the output. Default value is `NULL`.
`sig_level`	a number indicating the significance level to use in the General Procedure for Hypothesis Testing.
`na_rm`	a logical value indicating whether `NA` values should be removed before the computation proceeds.

Details

We have wrapped the `t.test` and the `BSDA::z.test` in a function as explained in the book of Montgomery and Runger (2010) <ISBN: 978-1-119-74635-5>.

Value

a `tibble` with the following columns:

statistic the value of the test statistic.

p_value the p-value for the test.

critical_value critical value in the General Procedure for Hypothesis Testing.

critical_region critical region in the General Procedure for Hypothesis Testing.

delta a scalar value indicating the value of Δ_0.

alternative character string giving the direction of the alternative hypothesis.

lower_ci lower bound of the confidence interval. It is presented only if `!is.null(conf_level)`.

upper_ci upper bound of the confidence interval. It is presented only if `!is.null(conf_level)`.

Examples

```
# t-test: var_equal == FALSE
x <- rnorm(1000, mean = 10, sd = 2)
y <- rnorm(500, mean = 5, sd = 1)
# H0: mu_1 - mu_2 == -1 versus H1: mu_1 - mu_2 != -1
ht_2pop_mean(x, y, delta = -1)
# t-test: var_equal == TRUE
x <- rnorm(1000, mean = 10, sd = 2)
y <- rnorm(500, mean = 5, sd = 2)
# H0: mu_1 - mu_2 == -1 versus H1: mu_1 - mu_2 != -1
ht_2pop_mean(x, y, delta = -1, var_equal = TRUE)

# z-test
x <- rnorm(1000, mean = 10, sd = 3)
x <- rnorm(500, mean = 5, sd = 1)
# H0: mu_1 - mu_2 >= 0 versus H1: mu_1 - mu_2 < 0
ht_2pop_mean(x, y, delta = 0, sd_pop_1 = 3, sd_pop_2 = 1)
```

ht_2pop_prop *Hypothesis testing for two population porportions*

Description

Comparing proportions in two populations

Usage

```
ht_2pop_prop(
  x,
  y,
  n_x = NULL,
  n_y = NULL,
  delta = 0,
  alternative = "two.sided",
  conf_level = NULL,
  sig_level = 0.05,
  na_rm = FALSE
)
```

Arguments

x	a vector of 0 and 1, or a scalar of count of sucesses in the first group.
y	a vector of 0 and 1, or a scalar of count of sucesses in the first group.
n_x	a scalar of number of trials in the first group.
n_y	a scalar of number of trials in the second group.
delta	a scalar value indicating the difference in proportions (Δ_0). Default value is 0.
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
conf_level	a number indicating the confidence level to compute the confidence interval. If conf_level = NULL, then the confidence interval is not included in the output. Default value is NULL.
sig_level	a number indicating the significance level to use in the General Procedure for Hypothesis Testing.
na_rm	a logical value indicating whether NA values should be removed before the computation proceeds. Default value is FALSE.

Details

ht_2pop_prop can be used for testing the null hipothesis that proportions (probabilities of success) in two groups are the same.

If is.null(n_x) == T and is.null(n_y) == T, then x and y must be a numeric value of 0 and 1 and the proportions are computed using x and y. If is.null(n_x) == F and is.null(n_y) == F, then x, y, n_x and n_y must be non-negative integer scalars and x <= n_x and y <= n_y.

Value

a `tibble` with the following columns:

statistic the value of the test statistic.

p_value the p-value for the test.

critical_value critical value in the General Procedure for Hypothesis Testing.

critical_region critical region in the General Procedure for Hypothesis Testing.

delta a scalar value indicating the value of `delta`.

alternative character string giving the direction of the alternative hypothesis.

lower_ci lower bound of the confidence interval. It is presented only if `!is.null(conf_level)`.

upper_ci upper bound of the confidence interval. It is presented only if `!is.null(conf_level)`.

Examples

```
x <- 3
n_x <- 100
y <- 50
n_y <- 333
ht_2pop_prop(x, y, n_x, n_y)

x <- rbinom(100, 1, 0.75)
y <- rbinom(500, 1, 0.75)
ht_2pop_prop(x, y)
```

`ht_2pop_var` *F Test to compare two variances*

Description

Performs a F test to compare the variances of two normal populations.

Usage

```
ht_2pop_var(
  x,
  y,
  ratio = 1,
  alternative = "two.sided",
  conf_level = FALSE,
  sig_level = 0.05,
  na_rm = FALSE
)
```

Arguments

x	a (non-empty) numeric vector.
y	a (non-empty) numeric vector.
ratio	the hypothesized ratio of the population variances of x and y. Default value is 1.
alternative	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
conf_level	a number indicating the confidence level to compute the confidence interval. If conf_level = NULL, then the confidence interval is not included in the output. Default value is NULL.
sig_level	a number indicating the significance level to use in the General Procedure for Hypothesis Testing.
na_rm	a logical value indicating whether NA values should be removed before the computation proceeds. Default value is FALSE.

Details

We have wrapped the var.test in a function as explained in the book of Montgomery and Runger (2010) <ISBN: 978-1-119-74635-5>.

Value

a tibble with the following columns:

statistic the value of the test statistic.

p_value the p-value for the test.

critical_value critical value in the General Procedure for Hypothesis Testing.

critical_region critical region in the General Procedure for Hypothesis Testing.

ratio a scalar value indicating the value of ratio.

alternative character string giving the direction of the alternative hypothesis.

lower_ci lower bound of the confidence interval. It is presented only if !is.null(conf_level).

upper_ci upper bound of the confidence interval. It is presented only if !is.null(conf_level).

Examples

```
x <- rnorm(100, sd = 2)
y <- rnorm(1000, sd = 10)
ht_2pop_var(x, y)
```

Index