

Package ‘biostats101’

May 7, 2026

Type Package

Title Practical Functions for Biostatistics Beginners

Version 0.1.1

Description A set of user-friendly functions designed to fill gaps in existing introductory biostatistics R tools, making it easier for newcomers to perform basic biostatistical analyses without needing advanced programming skills.

The methods implemented in this package are based on the works:

Connor (1987) <[doi:10.2307/2531961](https://doi.org/10.2307/2531961)>

Fleiss, Levin, & Paik (2013, ISBN:978-1-118-62561-3)

Levin & Chen (1999) <[doi:10.1080/00031305.1999.10474431](https://doi.org/10.1080/00031305.1999.10474431)>

McNemar (1947) <[doi:10.1007/BF02295996](https://doi.org/10.1007/BF02295996)>.

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr, ggplot2, tidyr

Suggests roxygen2, testthat

RoxygenNote 7.3.2

NeedsCompilation no

Author Marina Mautner Wizenier [aut, cre] (ORCID:

<<https://orcid.org/0000-0002-7259-5109>>),

Melody S. Goodman [aut] (ORCID:

<<https://orcid.org/0000-0001-8932-624X>>),

Jemar R. Bather [aut] (ORCID: <<https://orcid.org/0000-0002-0285-3678>>)

Maintainer Marina Mautner Wizenier <mm12025@nyu.edu>

Repository CRAN

Date/Publication 2024-09-12 18:40:02 UTC

Contents

lm_plot	2
mean_CI	3
power.2p.2n	3
power.paired.prop	5

lm_plot	<i>Create a Plot with Linear Regression and Confidence/Prediction Intervals</i>
---------	---

Description

Generates a ggplot2 plot that includes the observed data, a simple linear regression line, and both confidence and prediction interval bands.

Arguments

lm_model	A simple linear regression model object created by <code>lm()</code> .
conf.level	A numeric value indicating the confidence level for the confidence interval bands and prediction interval bands. Default is 0.95.
install_packages	Logical, indicating whether required packages should be automatically installed if not already available. Default is TRUE.

Value

A ggplot2 plot object featuring the observations, the simple linear regression line, the confidence interval bands, and the prediction interval bands.

Examples

```
# Example dataset
data <- data.frame(
  x = rnorm(100, mean = 50, sd = 10),
  y = 3 + 0.5 * rnorm(100, mean = 50, sd = 10) + rnorm(100)
)

# Run a regression model
my_model <- lm(y~x, data)

# Create plot with regression line, confidence limits, and prediction limits
lm_plot(my_model)

# Customize plot labels
lm_plot(my_model) + xlab("Your x-axis label") + ylab("Your y-axis label")
```

mean_CI	<i>Construct the Confidence Interval for the Mean</i>
---------	---

Description

Constructs a confidence interval (CI) for the mean of a numeric vector using the Student's t-distribution. The CI is calculated based on the specified degrees of freedom, confidence level, and alternative hypothesis.

Usage

```
mean_CI(data, conf.level = 0.95, alternative = "two.sided")
```

Arguments

data	A numeric vector from which the mean and confidence interval will be calculated.
conf.level	A numeric value representing the confidence level for the confidence interval estimation. Default is 0.95.
alternative	The alternative hypothesis to be considered on the confidence interval estimation. Options are 'two.sided' (default), 'greater', or 'less'.

Value

A named numeric vector with the mean and the lower and upper bounds of the confidence interval.

Examples

```
# Example data
values = c(5.2, 4.8, 6.3, 6.1, 7.2, 3.5, 4.9, 2.2, 3.7, 3.5, 8.9)

# Construct a 95% confidence interval for the mean
mean_CI(values)
```

power.2p.2n	<i>Calculate the Power and Sample Size for 2 Independent Proportions.</i>
-------------	---

Description

This function calculates

1. The power given independent proportions p_1 and p_2 , the sample sizes for the two groups, the confidence level and the alternative hypothesis.
2. The required sample size given independent proportions p_1 and p_2 , the desired power, the confidence level and the alternative hypothesis.

3. The required sample sizes n_1 and n_2 given the desired power, a sample size ratio n_2/n_1 (unbalanced designs), the confidence level and the alternative hypothesis.
4. The sample size for the second group given independent proportions p_1 and p_2 , the sample size for the first group (unbalanced designs), the desired power, the confidence level and the alternative hypothesis.

Usage

```
power.2p.2n(
  p1,
  p2,
  n1 = NULL,
  n2 = NULL,
  nratio = NULL,
  power = NULL,
  alternative = "two.sided",
  conf.level = 0.95,
  continuity = FALSE
)
```

Arguments

p1	Numeric, the proportion for the first group.
p2	Numeric, the proportion for the second group.
n1	Numeric, the sample size for the first group.
n2	Numeric, the sample size for the second group.
nratio	Numeric, the sample size ratio (n_2 / n_1) for unbalanced designs. Default is 1 when calculating sample sizes n_1 and n_2 .
power	Numeric, the desired power ($1 - \beta$). Default is 0.8 when calculating sample sizes n_1 and n_2 and when calculating n_2 given n_1 .
alternative	Character, the type of alternative hypothesis. Options are 'two.sided' (default) or 'one.sided'.
conf.level	Numeric, the confidence level ($1 - \alpha$). Default is 0.95.
continuity	Logical, indicating whether the continuity correction should be applied. Default is FALSE.

Value

A list with the following components:

- n: Total sample size ($n_1 + n_2$).
- n1: Sample size for the first group.
- n2: Sample size for the second group.
- power: The estimated power.
- nratio: The sample size ratio (n_2 / n_1), if applicable.

References

Levin, B., & Chen, X. (1999). Is the one-half continuity correction used once or twice to derive a well-known approximate sample size formula to compare two independent binomial distributions?. *The American Statistician*, 53(1), 62-66. <https://doi.org/10.1080/00031305.1999.10474431>.
 Fleiss, J. L., Levin, B., & Paik, M. C. (2013). *Statistical methods for rates and proportions*. John Wiley & Sons.

Examples

```
# Calculate the power for independent proportions given the sample sizes
power.2p.2n(p1 = 0.45, p2 = 0.6, n1 = 260, n2 = 130)

# Calculate the sample size for independent proportions (default power = 0.8)
power.2p.2n(p1 = 0.45, p2 = 0.6)

# Calculate n1 and n2 for independent proportions with ratio n2/n1
power.2p.2n(p1 = 0.44, p2 = 0.6, nratio = 2)

# Calculate n2 given n1 for independent proportions
power.2p.2n(p1 = 0.44, p2 = 0.6, n1 = 108)
```

power.paired.prop *Calculate the Power and Sample Size for Paired Proportions.*

Description

This function calculates either the power given the sample size or the sample size given the power for paired proportions p1 and p2.

Usage

```
power.paired.prop(
  p1,
  p2,
  n = NULL,
  power = NULL,
  conf.level = 0.95,
  alternative = "two.sided"
)
```

Arguments

p1	Numeric, the proportion at the first occasion.
p2	Numeric, the proportion at the second occasion.
n	Numeric, the sample size.
power	Numeric, the desired power (1 - beta). Default is 0.8 when calculating sample size.

conf.level	Numeric, the confidence level (1 - alpha). Default is 0.95.
alternative	Character, the type of alternative hypothesis. Options are 'two.sided' (default) or 'one.sided'.

Value

A list containing the sample size, power, confidence level, and alternative hypothesis.

References

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2), 153-157. <https://doi.org/10.1007/BF02295996>. Connor, R. J. (1987). Sample size for testing differences in proportions for the paired-sample design. *Biometrics*, 207-211. <https://doi.org/10.2307/2531961>.

Examples

```
# Calculate the power given the sample size for paired proportions  
power.paired.prop(p1 = 0.1, p2 = 0.15, n = 900)
```

```
# Calculate the sample size given the power for paired proportions  
power.paired.prop(p1 = 0.15, p2 = 0.1, power = 0.8)
```

Index

`lm_plot`, 2

`mean_CI`, 3

`power.2p.2n`, 3

`power.paired.prop`, 5