Package: biostats (via r-universe)

May 21, 2026

Title Biostatistics and Clinical Data Analysis

Version 1.1.2

Description Biostatistical and clinical data analysis, including descriptive statistics, exploratory data analysis, sample size and power calculations, statistical inference, and data visualization. Normality tests are implemented following Mishra et al. (2019) <doi:10.4103/aca.ACA_157_18>, omnibus test procedures are based on Blanca et al. (2017) <doi:10.3758/s13428-017-0918-2> and Field et al. (2012, ISBN:9781446200469), while sample size and power calculation methods follow Chow et al. (2017) <doi:10.1201/9781315183084>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0), covr

Config/testthat/edition 3

Imports ggplot2, gridExtra, rlang, car, gt, nortest

URL https://github.com/sebasquirarte/biostats

BugReports https://github.com/sebasquirarte/biostats/issues

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev

Repository https://sebasquirarte.r-universe.dev

Date/Publication 2026-04-21 17:28:49 UTC

RemoteUrl https://github.com/sebasquirarte/biostats

RemoteRef HEAD

RemoteSha ac6f9a18c68faad4f5dab18ab79645ca5ff5a978

Contents

`clinical_data` *Simulate Simple Clinical Trial Data*

Description

Creates a simple simulated clinical trial dataset with participant demographics, multiple visits, treatment groups with different effects, numerical and categorical variables, as well as optional missing data and dropout rates.

Usage

```
clinical_data(
  n = 100,
  visits = 3,
  arms = c("Placebo", "Treatment"),
  dropout = 0,
  missing = 0
)
```

Arguments

`n`	Integer indicating the number (1-999) of participants. Default: 100.
`visits`	Integer indicating the number of visits including baseline. Default: 3.
`arms`	Character vector of treatment arm names. Default: c("Placebo", "Treatment").
`dropout`	Numeric parameter indicating the proportion (0-1) of participants. who dropout. Default: 0.
`missing`	Numeric parameter indicating the proportion (0-1) of missing values to be introduced across numeric variables with fixed proportions (biomarker = 15%, weight = 25%, response = 60%). Default: 0.

Value

Dataframe with columns: participant_id, visit, sex, treatment, age, weight, biomarker, and response in long format.

Examples

```
# Basic dataset
clinical_df <- clinical_data()

# Multiple treatment arms with dropout rate and missing data
clinical_df <- clinical_data(arms = c('Placebo', 'A', 'B'), missing = 0.05, dropout = 0.10)
```

---

effect_measures *Effect Measures*

Description

Calculates measures of effect: Odds Ratio (OR), Risk Ratio (RR), and either Number Needed to Treat (NNT) or Number Needed to Harm (NNH).

Usage

```
effect_measures(
  exposed_event,
  exposed_no_event,
  unexposed_event,
  unexposed_no_event,
  alpha = 0.05,
  correction = TRUE
)

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

Arguments

exposed_event	Numeric value indicating the number of events in the exposed group.
exposed_no_event	Numeric value indicating the number of non-events in the exposed group.
unexposed_event	Numeric value indicating the number of events in the unexposed group.
unexposed_no_event	Numeric value indicating the number of non-events in the unexposed group.
alpha	Numeric value between 0 and 1 specifying the alpha level for confidence intervals (CI). Default: 0.05.

correction	Logical parameter that indicates whether a continuity correction (0.5) will be applied when any cell contains 0. Default: TRUE.
x	An object of class "effect_measures".
...	Further arguments passed to or from other methods.

Value

An object of class "effect_measures" containing the contingency table, effect size estimates (OR, RR, risk difference, NNT/NNH), and related statistics.

Methods (by generic)

- print(effect_measures): Print method for objects of class "effect_measures".

Examples

```
effect_measures(exposed_event = 15,
                exposed_no_event = 85,
                unexposed_event = 5,
                unexposed_no_event = 95)
```

---

missing_values *Descriptive and Visual Missing Value Assessment*

Description

Provides descriptive statistics and visualizations of missing values in a dataframe.

Usage

```
missing_values(data, color = "#79E1BE", all = FALSE)

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

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

Arguments

data	Dataframe containing the variables to be analyzed.
color	Character string indicating the color for missing values. Default: "#79E1BE"
all	Logical parameter that shows all variables including those without missing values. Default: FALSE.
x	An object of class "outliers".
...	Further arguments passed to or from other methods.

Value

An object of class "missing_values" containing descriptive statistics and ggplot objects.

Methods (by generic)

- `print(missing_values)`: Print method for objects of class "missing_values".

Functions

- `print(outliers)`: Print method for objects of class "outliers".

Examples

```
# Clinical dataset with missing values
clinical_df <- clinical_data(dropout = 0.1, missing = 0.05)

# Missing value analysis of only variables with missing values
missing_values(clinical_df)

# Show all variables including those without missing values
missing_values(clinical_df, all = TRUE)
```

`normality` *Statistical and Visual Normality Assessment*

Description

Tests normality using sample size-appropriate methods: Shapiro-Wilk test (n less than or equal to 50) or Kolmogorov-Smirnov test with Lilliefors' correction (n greater than 50) with Q-Q plots and histograms. Evaluates skewness and kurtosis using z-score criteria based on sample size. Automatically detects outliers and provides comprehensive visual and statistical assessment.

Usage

```
normality(data, x, all = FALSE, color = "#79E1BE")

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

Arguments

`data`	Dataframe containing the variables to be summarized.
`x`	An object of class "normality"
`all`	Logical parameter that displays all row indices of values outside 95% CI. Default: FALSE.
`color`	Character string indicating color for plots. Default: "#79E1BE".
`...`	Additional arguments (not used)

Value

Returns an object of class "normality" with normality statistics and ggplot objects.

Methods (by generic)

- `print(normality)`: Print method for objects of class "normality".

References

Mishra P., Pandey C.M., Singh U., Gupta A., Sahu C., and Keshri A. Descriptive statistics and normality tests for statistical data. Ann Card Anaesth. 2019 Jan-Mar;22(1):67-72. doi: 10.4103/aca.ACA_157_18. PMID: 30648682; PMCID: PMC6350423.

Lilliefors, H.W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. Journal of the American Statistical Association, 62(318), 399-402. doi: 10.1080/01621459.1967.10482916

Dallal, G.E. and Wilkinson, L. (1986). An analytic approximation to the distribution of Lilliefors' test for normality. The American Statistician, 40(4), 294-296. doi: 10.1080/00031305.1986.10475419

Examples

```
# Simulated clinical data
clinical_df <- clinical_data()

# Normally assesment of numerical variable
normality(clinical_df, "biomarker")

# Normally assesment of numerical variable with points outside 95% CI displayed
normality(clinical_df, "weight", all = TRUE)
```

---

omnibus *Omnibus Tests for Comparing Three or More Groups*

Description

Performs omnibus tests to evaluate overall differences between three or more groups. Automatically selects the appropriate statistical test based on data characteristics and assumption testing. Supports both independent groups and repeated measures designs. Tests include one-way ANOVA, repeated measures ANOVA, Kruskal-Wallis test, and Friedman test. Performs comprehensive assumption checking (normality, homogeneity of variance, sphericity) and post-hoc testing when significant results are detected.

Usage

```
omnibus(
  data,
  y,
  x,
```

```
  paired_by = NULL,
  alpha = 0.05,
  p_method = "holm",
  na.action = "na.omit"
)

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

Arguments

data	Dataframe containing the variables to be analyzed. Data must be in long format with one row per observation.
y	Character string indicating the dependent variable (outcome).
x	An object of class "omnibus".
paired_by	Character string indicating the source of repeated measurements. If provided, a repeated measures design is assumed. If NULL, independent groups design is assumed. Default: NULL.
alpha	Numeric value indicating the significance level for hypothesis tests. Default: 0.05.
p_method	Character string indicating the method for p-value adjustment in post-hoc multiple comparisons to control for Type I error inflation. Options: "holm" (Holm), "hochberg" (Hochberg), "hommel" (Hommel), "bonferroni" (Bonferroni), "BH" (Benjamini-Hochberg), "BY" (Benjamini-Yekutieli), "none" (no adjustment). Default: "holm".
na.action	Character string indicating the action to take if NAs are present ("na.omit" or "na.exclude"). Default: "na.omit"
...	Further arguments passed to or from other methods.

Value

An object of class "omnibus" containing the formula, statistic summary, name of the test performed, value of the test statistic, p value, alpha, the results of the post-hoc test and assumptions, the sample size's coefficient of variance, and corresponding degrees of freedom.

Methods (by generic)

- `print(omnibus)`: Print method for objects of class "omnibus".

References

Blanca, M., Alarcón, R., Arnau, J. et al. Effect of variance ratio on ANOVA robustness: Might 1.5 be the limit?. Behav Res. 2017 Jun 22; 50:937–962. https://doi.org/10.3758/s13428-017-0918-2 Field, A., Miles, J., & Field, Z. (2012). Discovering Statistics Using R. London: SAGE Publications.

Examples

```
# Simulated clinical data with multiple treatment arms and visits
clinical_df <- clinical_data(n = 300, visits = 6, arms = c("A", "B", "C"))

# Compare numerical variable across treatments
omnibus(data = clinical_df, y = "biomarker", x = "treatment")

# Filter simulated data to just one treatment
clinical_df_A <- clinical_df[clinical_df$treatment == "A", ]

# Compare numerical variable changes across visits
omnibus(y = "biomarker", x = "visit", data = clinical_df_A, paired_by = "participant_id")
```

---

outliers *Descriptive and Visual Outlier Assessment*

Description

Identifies outliers using Tukey's interquartile range (IQR) method and provides descriptive statistics and visualizations for outlier assessment in numeric data.

Usage

```
outliers(data, x, threshold = 1.5, color = "#79E1BE")
```

Arguments

data	Dataframe containing the variables to be analyzed.
x	Character string indicating the variable to be analyzed.
threshold	Numeric value multiplying the IQR to define outlier boundaries. Default: 1.5.
color	Character string indicating the color for non-outlier data points. Default: "#79E1BE".

Value

An object of class "outliers" containing a list with outlier statistics and ggplot objects.

Examples

```
# Simulated clinical data
clinical_df <- clinical_data()

# Basic outlier detection
outliers(clinical_df, "biomarker")

# Using custom threshold
outliers(clinical_df, "biomarker", threshold = 1.0)
```

plot_bar *Create Simple Professional Bar Plots*

Description

Generates publication-ready bar plots with minimal code using ggplot2.

Usage

```
plot_bar(
  data,
  x,
  y = NULL,
  group = NULL,
  facet = NULL,
  position = c("dodge", "stack", "fill"),
  stat = NULL,
  colors = NULL,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  legend_title = NULL,
  flip = FALSE,
  values = FALSE
)
```

Arguments

data	A data frame containing the variables to plot
x	Character string specifying the x-axis variable.
y	Character string specifying the y-axis variable. If NULL, counts calculated automatically. Default: NULL.
group	Character string specifying the grouping variable for fill color. Default: NULL.
facet	Character string specifying the faceting variable. Default: NULL.
position	Character string specifying bar position: "dodge", "stack", or "fill".
stat	Character string for statistical aggregation: "mean" or "median".
colors	Character vector of colors. If NULL, uses TealGrn palette. Default: NULL.
title	Character string for plot title. Default: NULL.
xlab	Character string for x-axis label. Default: NULL.
ylab	Character string for y-axis label. Default: NULL.
legend_title	Character string for legend title. Default: NULL.
flip	Logical parameter indicating whether to flip coordinates. Default: FALSE.
values	Logical parameter indicating whether to display value labels above bars. Default: FALSE.

Value

A ggplot2 object

Examples

```
# Simulated clinical data
clinical_df <- clinical_data()

# Proportion of response by treatment
plot_bar(data = clinical_df, x = "treatment", group = "response", position = "fill",
        title = "Proportion of response by treatment", values = TRUE)

# Grouped barplot of categorical variable by treatment with value labels
plot_bar(data = clinical_df, x = "response", group = "visit", facet = "treatment",
        title = "Response by visit and treatment",values = TRUE)
```

---

plot_box *Create Simple Professional Box Plots*

Description

Generates publication-ready boxplots with optional jittered points and mean overlay.

Usage

```
plot_box(
  data,
  x,
  y,
  group = NULL,
  facet = NULL,
  colors = NULL,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  legend_title = NULL,
  points = FALSE,
  point_size = 2,
  y_limits = NULL,
  show_mean = TRUE
)
```

Arguments

data	A dataframe containing the variables to plot.
x	Character string specifying the x-axis variable.

y	Character string specifying the y-axis variable.
group	Character string specifying grouping variable for fill/color. Default: NULL.
facet	Character string specifying faceting variable. Default: NULL.
colors	Character vector of colors. If NULL, uses TealGrn palette. Default: NULL.
title	Character string for plot title. Default: NULL.
xlab	Character string for x-axis label. Default: NULL.
ylab	Character string for y-axis label. Default: NULL.
legend_title	Character string for legend title. Default: NULL.
points	Logical parameter indicating if jittered points. Default: FALSE.
point_size	Numeric value indicating the size of points. Default: 2.
y_limits	Numeric vector of length 2 for y-axis limits. Default: NULL.
show_mean	Logical parameter indicating if mean should be shown. Default: TRUE.

Value

A ggplot2 object

Examples

```
#Simulated clinical data
clinical_df <- clinical_data(visits = 10)

# Boxplot of biomarker by treatment
plot_box(clinical_df, x = "treatment", y = "biomarker", group = "treatment")

# Barplot of bimarker by study visit and treatment
plot_box(clinical_df, x = "visit", y = "biomarker", group = "treatment")
```

---

plot_corr *Create Simple Professional Correlation Matrix Plots*

Description

Generates publication-ready correlation matrix heatmaps with minimal code using ggplot2.

Usage

```
plot_corr(
  data,
  vars = NULL,
  method = c("pearson", "spearman"),
  type = c("full", "upper", "lower"),
  colors = NULL,
  title = NULL,
```

```
  show_values = TRUE,
  value_size = 3,
  show_sig = FALSE,
  sig_level = 0.05,
  sig_only = FALSE,
  show_legend = TRUE,
  p_method = "holm"
)
```

Arguments

data	A dataframe containing the variables to analyze.
vars	Character vector specifying which variables to include. Default: NULL.
method	Character string specifying correlation method: "pearson" or "spearman". Default: "pearson".
type	Character string specifying matrix type: "full", "upper", or "lower". Default: "full".
colors	Character vector of 3 colors for negative, neutral, and positive correlations. Default: NULL.
title	Character string for plot title. Default: NULL.
show_values	Logical parameter indicating whether to display correlation values in cells. Default: TRUE.
value_size	Numeric value indicating size of correlation value text. Default: 3.
show_sig	Logical parameter indicating whether to mark significant correlations. Default: FALSE.
sig_level	Numeric value indicating significance level for marking. Default: 0.05.
sig_only	Logical parameter indicating whether to show only statistically significant values. Default: FALSE.
show_legend	Logical parameter indicating whether to show legend. Default: TRUE.
p_method	Character string specifying the method for p-value adjustment to control for multiple comparisons in correlation testing. Options: "holm" (Holm), "hochberg" (Hochberg), "hommel" (Hommel), "bonferroni" (Bonferroni), "BH" (Benjamini-Hochberg), "BY" (Benjamini-Yekutieli), or "none" (no adjustment). Default: "holm".

Value

A ggplot2 object

Examples

```
# Correlation matrix for base R dataset 'swiss'
plot_corr(data = swiss)

# Lower triangle with significance indicators and filtering
plot_corr(data = swiss, type = "lower", show_sig = TRUE, sig_only = TRUE)
```

plot_hist *Create Simple Professional Histogram Plots*

Description

Generates publication-ready histogram plots with minimal code using ggplot2.

Usage

```
plot_hist(
  data,
  x,
  group = NULL,
  facet = NULL,
  bins = 30,
  binwidth = NULL,
  alpha = 0.7,
  colors = NULL,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  legend_title = NULL,
  y_limits = NULL,
  x_limits = NULL,
  stat = NULL
)
```

Arguments

data	A dataframe containing the variables to plot.
x	Character string specifying the variable for the histogram.
group	Character string specifying the grouping variable for multiple histograms. Default: NULL.
facet	Character string specifying the faceting variable. Default: NULL.
bins	Numeric value indicating the number of bins for the histogram. Default: 30.
binwidth	Numeric value indicating the width of the bins (overrides bins if specified). Default: NULL.
alpha	Numeric value indicating the transparency level for the bars. Default: 0.7.
colors	Character vector of colors. If NULL, uses TealGrn palette. Default: NULL.
title	Character string for plot title. Default: NULL.
xlab	Character string for x-axis label. Default: NULL.
ylab	Character string for y-axis label. Default: NULL.
legend_title	Character string for legend title. Default: NULL.

y_limits	Numeric vector of length 2 for y-axis limits. Default: NULL.
x_limits	Numeric vector of length 2 for x-axis limits. Default: NULL.
stat	Character string that adds line for "mean" or "median". Default: NULL.

Value

A ggplot2 object

Examples

```
# Simulated clinical data
clinical_df <- clinical_data()

# Basic histogram
plot_hist(clinical_df, x = "biomarker")

# Grouped histogram
plot_hist(clinical_df, x = "biomarker", group = "treatment")

# Faceted histogram
plot_hist(clinical_df, x = "biomarker", facet = "treatment")
```

---

plot_line *Create Simple Professional Line Plots*

Description

Generates publication-ready line plots with minimal code using ggplot2.

Usage

```
plot_line(
  data,
  x,
  y,
  group = NULL,
  facet = NULL,
  stat = NULL,
  error = "se",
  error_width = 0.2,
  colors = NULL,
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  legend_title = NULL,
  points = TRUE,
  line_size = 1,
```

```
  point_size = 3,
  y_limits = NULL,
  x_limits = NULL
)
```

Arguments

data	A data frame containing the variables to plot.
x	Character string specifying the x-axis variable.
y	Character string specifying the y-axis variable.
group	Character string specifying the grouping variable for multiple lines. Default: NULL.
facet	Character string specifying the faceting variable. Default: NULL.
stat	Character string for statistical aggregation: "mean" or "median".
error	Character string for error bars: "se", "sd", "ci", or "none". Default: "se".
error_width	Numeric value indicating the width of error bar caps. Default: 0.2.
colors	Character vector of colors. If NULL, uses TealGrn palette. Default: NULL.
title	Character string for plot title. Default: NULL.
xlab	Character string for x-axis label. Default: NULL.
ylab	Character string for y-axis label. Default: NULL.
legend_title	Character string for legend title. Default: NULL.
points	Logical parameter indicating whether to add points to lines. Default: TRUE.
line_size	Numeric value indicating thickness of lines. Default: 1.
point_size	Numeric value indicating size of points if shown. Default: 3.
y_limits	Numeric vector of length 2 for y-axis limits. Default: NULL.
x_limits	Numeric vector of length 2 for x-axis limits. Default: NULL.

Value

A ggplot2 object

Examples

```
# Simulated clinical data
clinical_df <- clinical_data(arms = c("A","B","C"), visits = 10)

# Line plot with mean and standard error by treatment
plot_line(clinical_df, x = "visit", y = "biomarker",
          group = "treatment", stat = "mean", error = "se")

# Faceted line plots with median and 95% CI
plot_line(clinical_df, x = "visit", y = "biomarker", group = "treatment",
          facet = "sex", stat = "median", error = "ci", points = FALSE)
```

sample_size *Sample Size Calculation for Clinical Trials*

Description

Calculates the sample size needed in a clinical trial based on study design and statistical parameters using standard formulas for hypothesis testing (Chow, S. 2017).

Usage

```
sample_size(
  sample = c("one-sample", "two-sample"),
  design = NULL,
  outcome = c("mean", "proportion"),
  type = c("equality", "equivalence", "non-inferiority", "superiority"),
  alpha = 0.05,
  beta = 0.2,
  x1 = NULL,
  x2 = NULL,
  SD = NULL,
  delta = NULL,
  dropout = 0,
  k = 1
)

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

Arguments

sample	Character string indicating whether one or two samples need to be calculated. Options: "one-sample" or "two-sample".
design	Character string indicating study design when sample = "two-sample". Options: "parallel" or "crossover". Default: NULL for one-sample tests.
outcome	Character string indicating the type of outcome variable. Options: "mean" or "proportion".
type	Character string indicating the type of hypothesis test. Options: "equality", "equivalence", "non-inferiority", or "superiority".
alpha	Numeric parameter indicating the Type I error rate (significance level). Default: 0.05.
beta	Numeric parameter indicating the Type II error rate (1 - power). Default: 0.20.
x1	Numeric value of the mean or proportion for group 1 (treatment group).
x2	Numeric value of the mean or proportion for group 2 (control group or reference value).

SD	Numeric value indicating the standard deviation. Required for mean outcomes and crossover designs with proportion outcomes. Default: NULL.
delta	Numeric value indicating the margin of clinical interest. Required for non-equality tests. Must be negative for non-inferiority and positive for superiority/equivalence. Default: NULL.
dropout	Numeric value indicating the discontinuation rate expected in the study. Must be between 0 and 1. Default: 0.
k	Numeric value indicating the allocation ratio (n1/n2) for two-sample tests. Default: 1.
x	An object of class "sample_size".
...	Further arguments passed to or from other methods.

Value

An object of class "sample_size" containing the calculated sample size and study parameters.

Methods (by generic)

- `print(sample_size)`: Print method for objects of class "sample_size".

References

Chow, S.-C., Shao, J., Wang, H., & Lokhnygina, Y. (2017). Sample Size Calculations in Clinical Research (3rd ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781315183084

Examples

```
# Two-sample parallel non-inferiority test for means with 10% expected dropout
sample_size(sample = 'two-sample', design = 'parallel', outcome = 'mean',
            type = 'non-inferiority', x1 = 5.0, x2 = 5.0,
            SD = 0.1, delta = -0.05, k = 1, dropout = 0.1)

# One-sample equivalence test for means
sample_size(sample = "one-sample", outcome = "mean", type = "equivalence",
            x1 = 0, x2 = 0, SD = 0.1, delta = 0.05)
```

---

sample_size_range *Calculate and visualize sample size across a range of treatment effects*

Description

Calculates required sample sizes for specified power levels (70%, 80%, 90%) across a range of treatment effect values (`x1`), while keeping the control group value (`x2`) fixed. Internally calls `sample_size()` and generates a plot to visualize how total sample size changes with varying `x1`.

Usage

```
sample_size_range(x1_range, x2, step = 0.1, ...)

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

Arguments

x1_range	Numeric vector of length 2 specifying the range of values to evaluate for the treatment group mean or proportion (x1).
x2	Numeric value for the control group mean or proportion (reference value).
step	Numeric value indicating the step size to increment across the x1_range. Default: 0.1.
...	Further arguments passed to or from other methods.
x	An object of class "sample_size_range".

Value

An object of class "sample_size_range" containing the dataframe of sample size calculations and the ggplot object. A plot is also generated to visualize the relationship between treatment effects and required sample sizes.

Methods (by generic)

- print(sample_size_range): Print method for objects of class "sample_size_range".

References

Chow, S.-C., Shao, J., Wang, H., & Lokhnygina, Y. (2017). Sample Size Calculations in Clinical Research (3rd ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781315183084

See Also

sample_size

Examples

```
# Two-sample parallel non-inferiority test for proportions with 10% dropout
sample_size_range(x1_range = c(0.65, 0.75), x2 = 0.65, step = 0.01,
                  sample = "two-sample", design = "parallel", outcome = "proportion",
                  type = "non-inferiority", delta = -0.1, dropout = 0.1)

# One-sample equivalence test for means
sample_size_range(x1_range = c(-0.01, 0.01), x2 = 0, step = 0.005,
                  sample = "one-sample", outcome = "mean", type = "equivalence",
                  SD = 0.1, delta = 0.05, alpha = 0.05)
```

summary_table *Summary Table with Optional Group Comparisons*

Description

Generates a summary table for biostatistics and clinical data analysis with automatic normality, effect size, and statistical test calculations. Handles both numeric and categorical variables, performing appropriate descriptive statistics and inferential tests for single-group summaries or two-group comparisons.

Usage

```
summary_table(
  data,
  group_by = NULL,
  normality_test = "S-W",
  all = FALSE,
  effect_size = FALSE,
  exclude = NULL
)
```

Arguments

data	Dataframe containing the variables to be summarized.
group_by	Character string indicating the name of the grouping variable for two-group comparisons. Default: NULL.
normality_test	Character string indicating the normality test to use: 'S-W' for Shapiro-Wilk or 'K-S' for Kolmogorov-Smirnov with Lilliefors' correction. Default: 'S-W'.
all	Logical parameter that shows all calculated statistics. Default: FALSE.
effect_size	Logical parameter that includes effect size estimates. Default: FALSE.
exclude	Character vector of variable names to exclude from the summary. Default: NULL.

Value

A gt table object with formatted summary statistics.

Examples

```
# Simulated clinical data
clinical_df <- clinical_data()

# Overall summary without considering treatment groups
summary_table(clinical_df,
              exclude = c('participant_id', 'visit'))
```

```
# Grouped summary by treatment group
summary_table(clinical_df,
              group_by = 'treatment',
              exclude = c('participant_id', 'visit'))

# Grouped summary by treatment group with all stats and effect size
summary_table(clinical_df,
              group_by = 'treatment',
              all = TRUE,
              effect_size = TRUE,
              exclude = c('participant_id', 'visit'))
```

Index