

Package ‘dentomedical’

December 4, 2025

Type Package

Title Publication-Ready Descriptive, Bivariate, Regression, and Diagnostic Accuracy Tools for Medical and Dental Data

Version 0.1.3

Description The ‘dentomedical’ package provides a comprehensive suite of tools for medical and dental research. It includes automated descriptive statistics, bivariate analysis with intelligent test selection, logistic regression, and diagnostic accuracy assessment. All functions generate structured, publication-ready tables using ‘flextable’, ensuring reproducibility and clarity suitable for manuscripts, reports, and clinical research workflows.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, stats, flextable, tibble, MASS, broom, tidyr

Depends R (>= 4.0.0)

URL <https://github.com/umarhussain-git/dentomedical>

BugReports <https://github.com/umarhussain-git/dentomedical/issues>

NeedsCompilation no

Suggests testthat, knitr, rmarkdown

Author Umar Hussain [aut, cre]

Maintainer Umar Hussain <drumarhussain@gmail.com>

Repository CRAN

Date/Publication 2025-12-04 15:00:09 UTC

Contents

diag_accuracy	2
linreg	3
logreg	3
medical_data	4

2	<i>diag_accuracy</i>
norm_sum	4
sum_stat	5
sum_stat_p	5
Index	7

diag_accuracy	<i>Diagnostic Accuracy Metrics with Optional 2x2 Table</i>
----------------------	--

Description

Calculates diagnostic accuracy measures (Sensitivity, Specificity, PPV, NPV, Accuracy, LR+, LR-, DOR) from a binary test and gold standard. Provides 95% confidence intervals using Wilson method for proportions and log method for ratios. Optionally, prints a descriptive 2x2 table.

Usage

```
diag_accuracy(data, test_col, gold_col, descriptive = FALSE)
```

Arguments

data	A data frame containing the test results and gold standard.
test_col	Character. Name of the column in data with test results ("positive"/"negative").
gold_col	Character. Name of the column in data with gold standard results ("positive"/"negative").
descriptive	Logical. If TRUE, prints a descriptive 2x2 table with counts (TN, TP, FP, FN). Default is FALSE.

Value

A flextable object summarizing diagnostic metrics with 95% CI. If `descriptive = TRUE`, also prints a 2x2 table of counts.

Examples

```
diagnostic_data <- data.frame(
  test = c("positive", "negative", "positive", "negative", "positive", "negative", "positive", "negative"),
  goldstandard = c("positive", "positive", "negative", "negative", "positive", "negative", "positive", "negative")
)
diag_accuracy(diagnostic_data, test_col = "test",
              gold_col = "goldstandard",
              descriptive = FALSE)
```

linreg

*Linear Regression Summary Table***Description**

This function performs univariate and multivariate linear regression analyses for the specified predictors and outcome variable, returning a summary table with characteristics, regression coefficients (β) with 95% Numeric variables are summarized as mean (SD); categorical variables as n (\ Multivariate model R^2 and adjusted R^2 are included in the table footer.

Arguments

data	A data frame or tibble containing the variables.
outcome	The name of the outcome variable (numeric) as a string.
predictors	A character vector of predictor variable names.

Value

A flextable object summarizing univariate and multivariate linear regression results.

Examples

```
# Example using built-in iris dataset
linreg(iris, outcome = "Sepal.Length",
       predictors = c("Sepal.Width", "Petal.Length", "Species"))
```

logreg

*Logistic Regression Summary Table***Description**

Performs logistic regression for a binary outcome and a set of predictor variables. Computes both univariate and multivariate odds ratios (ORs) with 95% confidence intervals and p-values. Categorical variables automatically include a reference level in the output. Results are returned as a formatted flextable.

Arguments

data	A data frame containing the outcome and predictor variables.
outcome	A character string (factor) specifying the binary outcome variable.
predictors	A character vector (factor) of predictor variables to include in the regression.

Value

A flextable displaying univariate and multivariate odds ratios (95% CI) and p-values for each predictor. Reference levels for categorical variables are labeled "Reference".

Examples

```
logreg(data=medical_data(), outcome="case" ,
       predictors= c("age" , "parity" , "induced" ))
```

medical_data *Load Infertility Dataset*

Description

Load Infertility Dataset

Usage

```
medical_data()
```

Value

A data.frame containing infertility cases with labeled predictors suitable for logistic regression

norm_sum *Normality Test Summary Table for Numeric Variables*

Description

This function performs the Shapiro-Wilk normality test on all numeric variables in a dataset and returns the results in a publication-ready *flextable*. Extremely small p-values are displayed as "p < 0.001". The function automatically detects numeric variables and ignores non-numeric columns.

Arguments

- | | |
|--------------------|---|
| data | A data frame containing numeric and non-numeric variables. Only numeric variables are assessed for normality. |
| sample_size | Integer. Maximum number of observations to use for the Shapiro-Wilk test per variable (default = 5000). |

Value

A *flextable* summarizing each numeric variable with Shapiro-Wilk W statistic, formatted p-value, and distribution classification ("Normal" or "Skewed").

Examples

```
norm_sum(iris)
```

sum_stat*Descriptive Summary Table for Continuous and Categorical Variables*

Description

This function generates descriptive summary tables for both continuous and categorical variables. Continuous variables can be summarized using mean (SD) or median (IQR), and categorical variables are summarized as counts and percentages. Optionally, summaries can be stratified by a grouping variable.

Arguments

<code>data</code>	A data frame containing the variables to summarize.
<code>by</code>	Optional. A grouping variable (column name as string) to stratify the summary table.
<code>statistic</code>	Character. Summary statistic for continuous variables. Either " <code>mean_sd</code> " (default) or " <code>med_iqr</code> ".

Value

A `flextable` object displaying the summary table with appropriate formatting for publication or reporting. Continuous variables show mean (SD) or median (IQR), and categorical variables show counts and percentages. If `by` is specified, summaries are presented for each group in separate columns.

Examples

```
sum_stat(iris)
sum_stat(iris, by = "Species", statistic = "mean_sd")
sum_stat(iris, statistic = "med_iqr")
```

sum_stat_p*Create a Summary Table With P-Values for Group Comparisons*

Description

`sum_stat_p()` generates a descriptive summary table for both categorical and continuous variables stratified by a grouping variable. It automatically computes appropriate statistical tests (Chi-square, Fisher's exact, t-test, Wilcoxon, ANOVA, or Kruskal-Wallis) based on data type and distribution characteristics. The output is formatted as a `flextable` with footnotes indicating the summary statistics used and the tests applied.

Arguments

<code>data</code>	A data frame or tibble containing variables to summarise.
<code>by</code>	A string specifying the grouping variable name. Must be a column in <code>data</code> .
<code>statistic</code>	A string specifying summary style for continuous variables: <ul style="list-style-type: none"> • <code>"mean_sd"</code>: Mean (SD) • <code>"med_iqr"</code>: Median (IQR)
<code>test_type</code>	Optionally force a specific test. Choices: <ul style="list-style-type: none"> • <code>"auto"</code> (<i>default</i>) — automatically selects appropriate tests • <code>"chisq"</code>, <code>"fisher"</code> for categorical variables • <code>"t.test"</code>, <code>"wilcox"</code> for 2-group continuous comparisons • <code>"anova"</code>, <code>"kruskal"</code> for >2-group continuous comparisons

Value

A flextable object containing the summary table with p-values and footer notes describing summary statistics and tests used.

Examples

```
# Load built-in dataset
data(CO2)

# Example 1: Auto test selection, median/IQR summary
sum_stat_p(CO2, by = "Type", statistic = "med_iqr")

# Example 2: Force Wilcoxon test for continuous variables
sum_stat_p(CO2, by = "Type", statistic = "med_iqr", test_type = "wilcox")

# Example 3: Mean/SD with automatic test choice
sum_stat_p(CO2, by = "Treatment", statistic = "mean_sd")
```

Index

diag_accuracy, 2

linreg, 3

logreg, 3

medical_data, 4

norm_sum, 4

sum_stat, 5

sum_stat_p, 5