Package: dentomedical (via r-universe)

May 13, 2026

Type Package

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

Version 0.1.0

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

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

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

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev

Repository https://umarhussain-git.r-universe.dev

Date/Publication 2026-01-13 05:51:13 UTC

RemoteUrl https://github.com/umarhussain-git/dentomedical

RemoteRef HEAD

RemoteSha fe7c9a2e2be2262badc9c1bb94fbe1ed95d5ce13

Contents

`diag_accuracy` *Diagnostic Accuracy Metrics with Optional 2x2 Table*

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 = TRUE)
```

`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 ( ) with 95% CI, and p-values. Numeric variables show mean (SD); categorical variables show n (%). Multivariate model R^2 and adjusted R^2 are included in the table footer.

Usage

```
linreg(data, outcome, predictors)
```

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.

Usage

```
logreg(data, outcome, predictors)
```

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.

Usage

```
norm.sum(data, sample_size = 5000)
```

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.

Usage

```
## S3 method for class 'stat'
sum(data, by = NULL, statistic = "mean_sd")
```

Arguments

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

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.

Usage

```
## S3 method for class 'stat.p'
sum(data, by, statistic = "mean_sd", test_type = "auto")
```

Arguments

`data`	A data frame or tibble containing variables to summarise.
`by`	A string specifying the grouping variable name. Must be a column in `data`.
`statistic`	A string specifying summary style for continuous variables: • `"mean_sd"`: Mean (SD) • `"med_iqr"`: Median (IQR)
`test_type`	Optionally force a specific test. Choices: • `"auto"` *(default)* — automatically selects appropriate tests • `"chisq"`, `"fisher"` for categorical variables • `"t.test"`, `"wilcox"` for 2-group continuous comparisons • `"anova"`, `"kruskal"` 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