

Package ‘SimtablR’

February 16, 2026

Title Easy Publication-Ready Tables and Regression Analysis

Version 1.0.1

Description Streamlines the creation of descriptive frequency tables ('Table 1'), diagnostic test accuracy evaluations (sensitivity, specificity, predictive values), and multi-outcome regression summaries. Features automatic tables, prevalence and odds ratio calculations, and seamless integration with 'flextable' for exporting results to 'Microsoft Word' and 'PowerPoint'.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports cli, dplyr, flextable, lmtest, openxlsx, sandwich, stats, tidy, utils

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Depends R (>= 3.5)

LazyData true

URL <https://github.com/MatheusTG-14/SimtablR>,
<https://MatheusTG-14.github.io/SimtablR/>

BugReports <https://github.com/MatheusTG-14/SimtablR/issues>

NeedsCompilation no

Author Matheus Trabuco Gonzalez [aut, cre]

Maintainer Matheus Trabuco Gonzalez <matheustrabucogonzalez@gmail.com>

Repository CRAN

Date/Publication 2026-02-16 17:50:07 UTC

Contents

as.data.frame.diag_test	2
as.data.frame.tb	3

as_flextable.tb	3
diag_test	4
epitabl	5
export_regtab_csv	6
export_regtab_xlsx	6
plot.diag_test	7
print.diag_test	7
print.regtab	8
print.tb	8
regtab	9
tb	12
Index	14

as.data.frame.diag_test
<i>Convert diag_test Object to Data Frame</i>

Description

Convert diag_test Object to Data Frame

Usage

```
## S3 method for class 'diag_test'
as.data.frame(x, ...)
```

Arguments

- x A diag_test object
- ... Additional arguments (unused)

Value

A data.frame containing the performance metrics

as.data.frame.tb	<i>Convert tb to Data Frame</i>
------------------	---------------------------------

Description

Extracts the data from a tb object as a standard data.frame.

Usage

```
## S3 method for class 'tb'  
as.data.frame(x, ...)
```

Arguments

x	A tb object.
...	Additional arguments.

Value

A data.frame with the formatted table.

as_flextable.tb	<i>Convert tb object to flextable</i>
-----------------	---------------------------------------

Description

Converts a tb object into a flextable for high-quality reporting in Word or PowerPoint.

Usage

```
## S3 method for class 'tb'  
as_flextable(x, ...)
```

Arguments

x	A tb object.
...	Additional arguments passed to flextable::flextable.

Value

A flextable object ready for printing or exporting.

diag_test

*Diagnostic Test Accuracy Assessment***Description**

Computes a comprehensive confusion matrix and diagnostic performance metrics for binary classification tests.

Usage

```
diag_test(
  data,
  test,
  ref,
  positive = NULL,
  test_positive = NULL,
  conf.level = 0.95
)
```

Arguments

data	A data.frame containing the test and reference variables.
test	Unquoted name of the diagnostic test variable (binary).
ref	Unquoted name of the reference standard variable (binary).
positive	Character or numeric. Level representing "Positive" in REFERENCE.
test_positive	Character or numeric. Level representing "Positive" in TEST.
conf.level	Numeric. Confidence level (0-1). Default: 0.95.

Details

Confusion Matrix Structure The function creates a 2x2 confusion matrix:

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

Metrics Calculated

- Sensitivity, Specificity, PPV, NPV
- Accuracy, Prevalence
- Likelihood Ratios, Youden's Index, F1 Score

Value

An object of class `diag_test` containing:

- `table`: 2x2 confusion matrix
- `stats`: Data frame with metrics and CIs
- `labels`: List with labels used
- `sample_size`: Total valid observations

`epitabl`*Simulated Epidemiological Dataset*

Description

A simulated dataset containing demographic, clinical, and outcome variables for 500 individuals. Designed for demonstrating table creation and diagnostic testing analysis.

Usage

```
epitabl
```

Format

A data frame with 500 rows and 19 variables:

id Unique patient identifier

age Age in years (Numeric)

sex Biological sex (Female, Male)

bmi Body Mass Index in kg/m2 (Numeric, contains NAs)

smoking Smoking status (Never, Former, Current)

exercise Physical activity level (Low, Moderate, High)

education Educational attainment (High School, Some College, College+)

income Annual household income (<30k, 30-60k, 60k+)

disease Disease status - primary outcome (No, Yes)

rapid_test Result of rapid diagnostic test (Negative, Positive)

lab_confirmed Laboratory confirmation - gold standard (No, Yes)

comorbidity_score Score 0-5 based on medical history

outcome1 Count of primary care visits in past year

outcome2 Count of specialist visits in past year

outcome3 Count of emergency department visits in past year

hospitalized Hospitalized in past year (No, Yes)

systolic_bp Systolic blood pressure in mmHg

cholesterol Total cholesterol in mg/dL

region Geographic region (North, South, East, West)

Source

Simulated data for the SimtabLR package.

Examples

```
data(epitabl)

# Basic description
tb(epitabl, sex, disease)
```

export_regtab_csv	<i>Export regtab Results to CSV</i>
-------------------	-------------------------------------

Description

Export regtab Results to CSV

Usage

```
export_regtab_csv(x, file, ...)
```

Arguments

x	A data.frame from regtab
file	File path for export
...	Additional arguments passed to write.csv

Value

No return value, called for side effects

export_regtab_xlsx	<i>Export regtab Results to Excel</i>
--------------------	---------------------------------------

Description

Requires the 'openxlsx' package

Usage

```
export_regtab_xlsx(x, file, ...)
```

Arguments

x	A data.frame from regtab
file	File path for export (.xlsx)
...	Additional arguments passed to openxlsx::write.xlsx

Value

No return value, called for side effects

plot.diag_test	<i>Plot Diagnostic Test Results</i>
----------------	-------------------------------------

Description

Visualizes the confusion matrix using a fourfold plot.

Usage

```
## S3 method for class 'diag_test'  
plot(x, col = c("#ffcccc", "#ccffcc"), main = "Confusion Matrix", ...)
```

Arguments

x	A diag_test object.
col	Vector of 2 colors for Negative and Positive outcomes.
main	Title of the plot.
...	Additional arguments passed to fourfoldplot.

Value

No return value, called for side effects

print.diag_test	<i>Print Method for diag_test Objects</i>
-----------------	---

Description

Print Method for diag_test Objects

Usage

```
## S3 method for class 'diag_test'  
print(x, digits = 3, ...)
```

Arguments

- x A diag_test object
- digits Number of decimal places for metrics (default: 3)
- ... Additional arguments (unused)

Value

No return value, called for side effects

print.regtab	<i>Print Method for regtab Results</i>
--------------	--

Description

Print Method for regtab Results

Usage

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

Arguments

- x A data.frame returned by regtab
- ... Additional arguments passed to print

Value

No return value, called for side effects

print.tb	<i>Print Method for tb Objects</i>
----------	------------------------------------

Description

Prints a formatted table with flags and attributes.

Usage

```
## S3 method for class 'tb'  
print(x, digits = NULL, ...)
```

Arguments

<code>x</code>	A tb object.
<code>digits</code>	Number of decimal places to display.
<code>...</code>	Additional arguments.

Value

No return value, called for side effects

regtab	<i>Multi-Outcome Regression Table</i>
--------	---------------------------------------

Description

Fits generalized linear models (GLMs) for multiple outcome variables and generates a formatted wide-format table with point estimates and confidence intervals. Supports robust standard errors, automatic exponentiation for count/binary outcomes, and custom labeling for publication-ready tables.

Usage

```
regtab(
  data,
  outcomes,
  predictors,
  family = poisson(link = "log"),
  robust = TRUE,
  exponentiate = NULL,
  labels = NULL,
  d = 2,
  conf.level = 0.95,
  include_intercept = FALSE,
  p_values = FALSE
)
```

Arguments

<code>data</code>	Data.frame containing all variables for analysis.
<code>outcomes</code>	Character vector of dependent variable names. Each outcome is modeled separately with the same set of predictors.
<code>predictors</code>	Formula or character string specifying predictors. Can be: • Formula: <code>~ x1 + x2 + x3</code> • Character: <code>"~ x1 + x2 + x3"</code> or <code>"x1 + x2 + x3"</code>
<code>family</code>	GLM family specification. Options:

	• <code>poisson(link = "log")</code> - For count outcomes (default) • <code>binomial(link = "logit")</code> - For binary outcomes • <code>gaussian(link = "identity")</code> - For continuous outcomes • <code>quasipoisson()</code>, <code>quasibinomial()</code> - For overdispersed data • Or character: "poisson", "binomial", "gaussian"
<code>robust</code>	Logical. If TRUE (default), calculates heteroskedasticity-consistent (HC0) robust standard errors via the sandwich package. CIs are based on robust SEs.
<code>exponentiate</code>	Logical. If TRUE, exponentiates coefficients and CIs: • Poisson: IRR (Incidence Rate Ratios) • Binomial: OR (Odds Ratios) • Gaussian: Not typically used (stays on linear scale) If NULL (default), automatically detects: TRUE for Poisson/Binomial, FALSE for Gaussian.
<code>labels</code>	Named character vector for renaming outcome columns in output. Format: <code>c("raw_name" = "Pretty Label")</code> . Useful for publication tables.
<code>d</code>	Integer. Number of decimal places for rounding estimates and CIs. Default: 2.
<code>conf.level</code>	Numeric. Confidence level for intervals (0-1). Default: 0.95.
<code>include_intercept</code>	Logical. If TRUE, includes intercept in output table. Default: FALSE (typically excluded from publication tables).
<code>p_values</code>	Logical. If TRUE, adds p-values as separate column. Default: FALSE.

Details

Model Fitting:

For each outcome, the function fits: `glm(outcome ~ predictors, family = family, data = data)`

Robust Standard Errors:

When `robust = TRUE`, the function:

1. Fits the model with standard GLM.
2. Computes sandwich covariance matrix (HC0 estimator).
3. Calculates Wald-type CIs based on robust SEs.

This provides protection against heteroskedasticity and mild model misspecification.

Exponentiation:

- **Poisson regression:** $\exp(\beta) = \text{Incidence Rate Ratio}$
 - $\text{IRR} = 1$: No association
 - $\text{IRR} > 1$: Increased rate
 - $\text{IRR} < 1$: Decreased rate
- **Logistic regression:** $\exp(\beta) = \text{Odds Ratio}$
 - $\text{OR} = 1$: No association
 - $\text{OR} > 1$: Increased odds
 - $\text{OR} < 1$: Decreased odds

Output Format:

Returns a wide-format data.frame:

Variable	Outcome1	Outcome2	...
(Intercept)	2.34 (1.89-2.91)	1.98 (1.65-2.38)	...
age	1.05 (1.02-1.08)	1.03 (1.01-1.06)	...
sex	0.87 (0.75-1.01)	0.92 (0.81-1.05)	...

Each cell contains: "Estimate (Lower CI - Upper CI)"

Missing Data:

GLM uses complete cases by default. Observations with missing values in any variable are excluded from that specific model.

Convergence Issues:

If a model fails to converge or encounters errors:

- A warning is issued with the outcome name and error message
- That outcome column is skipped in the output
- Other outcomes continue processing

Value

A data.frame in wide format with:

- **Variable:** Predictor names (first column)
- **Outcome columns:** One column per outcome with formatted estimates and CIs

Can be directly exported to Excel, Word, or LaTeX for publication.

Examples

```
# Create example data
set.seed(456)
n <- 500
df <- data.frame(
  age = rnorm(n, 50, 10),
  sex = factor(sample(c("M", "F"), n, replace = TRUE)),
  treatment = factor(sample(c("A", "B"), n, replace = TRUE)),
  outcome1 = rpois(n, lambda = 5),
  outcome2 = rpois(n, lambda = 8),
  outcome3 = rpois(n, lambda = 3)
)

# Basic usage: Poisson regression for multiple outcomes
regtab(df,
  outcomes = c("outcome1", "outcome2", "outcome3"),
  predictors = ~ age + sex + treatment,
  family = poisson(link = "log"))

# With custom labels and no robust SEs
```

```
regtab(df,
  outcomes = c("outcome1", "outcome2"),
  predictors = "age + sex",
  labels = c(outcome1 = "Primary Endpoint", outcome2 = "Secondary Endpoint"),
  robust = FALSE)

# Logistic regression with p-values
df$binary_outcome <- rbinom(n, 1, 0.4)
regtab(df,
  outcomes = "binary_outcome",
  predictors = ~ age + sex,
  family = binomial(),
  p_values = TRUE)
```

tb

Frequency and Summary Tables

Description

Creates comprehensive tables for categorical or continuous variables with formatting, statistical tests, prevalence ratios (PR), odds ratios (OR), and column stratification. Supports both univariate and bivariate analysis with flexible output formatting.

Usage

```
tb(
  data,
  ...,
  m = FALSE,
  d = 1,
  format = TRUE,
  style = "n_pct",
  style.rp = "{rp} ({lower} - {upper})",
  style.or = "{or} ({lower} - {upper})",
  test = FALSE,
  subset = NULL,
  strat = NULL,
  rp = FALSE,
  or = FALSE,
  ref = NULL,
  conf.level = 0.95,
  var.type = NULL,
  stat.cont = "median"
)
```

Arguments

<code>data</code>	A data.frame or atomic vector.
<code>...</code>	Variables to be tabulated. Can include variable names and/or flags (m, p, row, col, rp, or) for controlling output format.
<code>m</code>	Logical. If TRUE, includes missing values (NA) in the table. Default: FALSE.
<code>d</code>	Integer. Number of decimal places for percentages and statistics. Default: 1.
<code>format</code>	Logical. If TRUE, renders a formatted grid output (Stata-style). If FALSE, prints as simple matrix. Default: TRUE.
<code>style</code>	Character. Format for displaying counts and percentages. Options: • "n_pct": "100 (25.0\ • "pct_n": "25.0\ • Custom template using {n} and {p} placeholders, e.g., "{n} [{p}%]"
<code>style.rp</code>	Character. Format string for Prevalence Ratio. Default: "{rp} ({lower} - {upper})".
<code>style.or</code>	Character. Format string for Odds Ratio. Default: "{or} ({lower} - {upper})".
<code>test</code>	Logical or Character. Performs statistical test on 2x2+ tables. • TRUE: Automatic selection (chi-squared for categorical) • "chisq", "fisher", "mcnemar"
<code>subset</code>	Logical expression for row filtering.
<code>strat</code>	Variable for column stratification. Disables PR/OR calculations.
<code>rp</code>	Logical. If TRUE, calculates Prevalence Ratios (PR).
<code>or</code>	Logical. If TRUE, calculates Odds Ratios (OR).
<code>ref</code>	Character or numeric. Specifies the reference level for PR/OR calculations.
<code>conf.level</code>	Numeric. Confidence level for intervals (0-1). Default: 0.95.
<code>var.type</code>	Named character vector specifying variable types. • Select the line variable and choose between "continuous" and "cont". • Example: <code>tb(epitabl, age, sex, var.type = c(age = "continuous"))</code>
<code>stat.cont</code>	Character. "mean" (Mean/SD) or "median" (Median/IQR).

Value

An object of class `tb` which is a matrix with attributes containing statistics, formatted flags, and class information.

Index

* **datasets**

epitabl, [5](#)

as.data.frame.diag_test, [2](#)

as.data.frame.tb, [3](#)

as_flextable.tb, [3](#)

diag_test, [4](#)

epitabl, [5](#)

export_regtab_csv, [6](#)

export_regtab_xlsx, [6](#)

plot.diag_test, [7](#)

print.diag_test, [7](#)

print.regtab, [8](#)

print.tb, [8](#)

regtab, [9](#)

tb, [12](#)