Package: SimtablR (via r-universe)

May 21, 2026

Title Easy Publication-Ready Tables and Regression Analysis

Version 1.2.0

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

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

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

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 4.1.0)

LazyData true

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

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

Config/testthat/edition 3

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://matheustg-14.r-universe.dev

Date/Publication 2026-02-20 22:32:54 UTC

RemoteUrl https://github.com/matheustg-14/simtablr

RemoteRef HEAD

RemoteSha a79eb73c628631d62d109ef7069a3e70c208ff6a

Contents

`as.data.frame.diag_test` *Convert diag_test to Data Frame*

Description

Extracts the performance metrics table as a plain `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` with columns `Metric`, `Estimate`, `LowerCI`, `UpperCI`.

as.data.frame.tb *Convert tb to Data Frame*

Description

Convert tb to Data Frame

Usage

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

Arguments

x	A tb object.
...	Additional arguments (unused).

Value

A data.frame with the formatted table.

as_flextable.tb *Convert tb Object to Flextable*

Description

Convert tb Object to Flextable

Usage

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

Arguments

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

Value

A flextable object.

`diag_test` *Diagnostic Test Accuracy Assessment*

Description

Computes a 2x2 confusion matrix and comprehensive diagnostic performance metrics for a binary classification test, with exact binomial confidence intervals.

Usage

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

Arguments

`data`	A data.frame containing `test` and `ref` variables.
`test`	Unquoted name of the diagnostic test variable (must be binary).
`ref`	Unquoted name of the reference standard variable (must be binary).
`positive`	Character or numeric. Level representing "Positive" in the **reference** variable. If `NULL` (default), auto-detected from common positive labels (`"Yes"`, `"1"`, `"Positive"`, etc.) or the last level.
`test_positive`	Character or numeric. Level representing "Positive" in the **test** variable. If `NULL` (default), mirrors `positive` when the same label exists in the test variable, then falls back to auto-detection.
`conf.level`	Numeric. Confidence level for binomial CIs (0-1). Default: `0.95`.

Details

Confusion Matrix Layout:

```
           | Ref +   | Ref -
-----------+---------+--------
Test +     |   TP    |   FP
Test -     |   FN    |   TN
```

Metrics Computed:

- **Sensitivity** (Recall) = TP / (TP + FN)
- **Specificity** = TN / (TN + FP)
- **PPV** (Precision) = TP / (TP + FP)

- **NPV** = TN / (TN + FN)
- **Accuracy** = (TP + TN) / Total
- **Prevalence** = (TP + FN) / Total
- **Likelihood Ratio +** = Sensitivity / (1 - Specificity)
- **Likelihood Ratio -** = (1 - Sensitivity) / Specificity
- **Youden's Index** = Sensitivity + Specificity - 1
- **F1 Score** = 2 x (PPV x Sensitivity) / (PPV + Sensitivity)

Binomial CIs (exact Clopper-Pearson) are computed for the first six metrics. Likelihood Ratios, Youden's Index, and F1 Score do not have CIs.

Value

An object of class `diag_test` - a named list with:

- `$table`: 2x2 `table` object (Test x Ref).
- `$stats`: `data.frame` with columns `Metric`, `Estimate`, `LowerCI`, `UpperCI`.
- `$labels`: named list with `ref_pos`, `ref_neg`, `test_pos`, `test_neg`.
- `$sample_size`: integer, total valid observations.
- `$conf.level`: numeric, confidence level used.

See Also

`print.diag_test()`, `as.data.frame.diag_test()`, `plot.diag_test()`

Examples

```
set.seed(1)
n   <- 200
ref <- factor(sample(c("No", "Yes"), n, replace = TRUE, prob = c(.55, .45)))
tst <- ifelse(ref == "Yes",
              ifelse(runif(n) < .80, "Yes", "No"),
              ifelse(runif(n) < .85, "No",  "Yes"))
df  <- data.frame(rapid_test = factor(tst), lab = ref)

result <- diag_test(df, test = rapid_test, ref = lab,
                    positive = "Yes", test_positive = "Yes")
print(result)
as.data.frame(result)
```

`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 *Export regtab Results to CSV*

Description

Export regtab Results to CSV

Usage

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

Arguments

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

Value

Invisibly returns x.

export_regtab_xlsx *Export regtab Results to Excel*

Description

Requires the openxlsx package.

Usage

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

Arguments

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

Value

Invisibly returns x.

plot.diag_test *Plot Diagnostic Test Results*

Description

Draws a fourfold display of the confusion matrix with sensitivity and specificity annotated on the bottom margin.

Usage

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

Arguments

x	A diag_test object.
col	Character vector of length 2. Fill colours for the negative and positive quadrants respectively. Default: c("#ffcccc", "#ccffcc").
main	Character. Plot title. Default: "Confusion Matrix".
...	Additional arguments passed to graphics::fourfoldplot().

Value

Invisibly returns x.

print.diag_test *Print Method for diag_test Objects*

Description

Displays a formatted summary of the confusion matrix and all diagnostic performance metrics with confidence intervals.

Usage

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

Arguments

x	A diag_test object.
digits	Integer. Decimal places for metrics. Default: 3.
...	Additional arguments (unused).

Value

Invisibly returns x.

print.regtab *Print Method for regtab Results*

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

Invisibly returns x.

print.tb *Print Method for tb Objects*

Description

Print Method for tb Objects

Usage

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

Arguments

x	A tb object.
digits	Number of decimal places to display.
...	Additional arguments (unused).

Value

Invisibly returns x, called for side effects.

regtab *Multi-Outcome Regression Table*

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

data	Data.frame containing all variables for analysis.
outcomes	Character vector of dependent variable names. Each outcome is modeled separately with the same set of predictors.
predictors	Formula or character string specifying predictors. Can be: • Formula: ~ x1 + x2 + x3 • Character: "~ x1 + x2 + x3" or "x1 + x2 + x3"
family	GLM family specification. Options: • poisson(link = "log") - For count outcomes (default) • binomial(link = "logit") - For binary outcomes • gaussian(link = "identity") - For continuous outcomes • quasipoisson(), quasibinomial() - For overdispersed data • Or character: "poisson", "binomial", "gaussian"
robust	Logical. If TRUE (default), calculates heteroskedasticity-consistent (HC0) robust standard errors via the sandwich package. CIs are based on robust SEs.
exponentiate	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.

`labels` Named character vector for renaming outcome columns in output. Format: `c("raw_name" = "Pretty Label")`. Useful for publication tables.

`d` Integer. Number of decimal places for rounding estimates and CIs. Default: 2.

`conf.level` Numeric. Confidence level for intervals (0-1). Default: 0.95.

`include_intercept`
Logical. If TRUE, includes intercept in output table. Default: FALSE (typically excluded from publication tables).

`p_values` 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) = Incidence Rate Ratio
  - IRR = 1: No association
  - IRR > 1: Increased rate
  - IRR < 1: Decreased rate
- **Logistic regression**: exp(beta) = Odds Ratio
  - OR = 1: No association
  - OR > 1: Increased odds
  - 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.

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

data	A data.frame or atomic vector.
...	Variables to be tabulated. Accepts variable names and/or flags (m, p, row, col, rp, or) for controlling output format.
m	Logical. Include missing values (NA) in the table. Default: FALSE.
d	Integer. Decimal places for percentages and statistics. Default: 1.
format	Logical. Render a formatted grid output. Default: TRUE.
style	Character. Format for displaying counts and percentages. Options: "n_pct" (default), "pct_n", or a custom template with {n} and {p} placeholders, e.g. "{n} [{p}%]".

style.rp	Character. Format string for Prevalence Ratio. Default: "{rp} ({lower} - {upper})".
style.or	Character. Format string for Odds Ratio. Default: "{or} ({lower} - {upper})".
test	Logical or Character. Performs statistical test on 2x2+ tables. TRUE for automatic selection, or one of "chisq", "fisher", "mcnemar".
subset	Logical expression for row filtering.
strat	Variable for column stratification. Disables PR/OR calculations.
rp	Logical. Calculate Prevalence Ratios (PR). Default: FALSE.
or	Logical. Calculate Odds Ratios (OR). Default: FALSE.
ref	Character or numeric. Reference level for PR/OR calculations.
conf.level	Numeric. Confidence level for intervals (0-1). Default: 0.95.
var.type	Named character vector specifying variable types, e.g. c(age = "continuous").
stat.cont	Character. "mean" (Mean/SD) or "median" (Median/IQR). Default: "median".

Value

An object of class tb (a matrix with attributes).

Index