Package 'gtregression'

August 18, 2025

Type Package

Title Tools for Creating Publication-Ready Regression Tables

Version 1.0.0

Description Simplifies regression modeling in R by integrating multiple modeling and summarization tools into a cohesive, user-friendly interface. Designed to be accessible for researchers, particularly those in Low- and Middle-Income Countries (LMIC). Built upon widely accepted statistical methods, including logistic regression (Hosmer et al. 2013, ISBN:9781118548429), log-binomial regression (Spiegelman and Hertzmark 2005 <doi:10.1093/aje/kwi188>), Poisson and robust Poisson regression (Zou 2004 <doi:10.1093/aje/kwh090>), negative binomial regression (Hilbe 2011, ISBN:9780521179515), and linear regression (Kutner et al. 2005, ISBN:9780071122214). Leverages multiple dependencies to ensure high-quality output and generate reproducible, publication-ready tables in alignment with best practices in epidemiology and applied statistics.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports dplyr, gtsummary, risks, purrr, MASS, rlang, stats, lmtest, patchwork, ggtext, ggplot2, tidyr, utils, sandwich, tibble, broom, broom.helpers, gt, officer, flextable

VignetteBuilder knitr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, mlbench, car, forcats, pkgdown

RoxygenNote 7.3.2

Config/testthat/edition 3

URL https://thinkdenominator.github.io/gtregression/

BugReports https://github.com/ThinkDenominator/gtregression/issues

NeedsCompilation no

Author Rubeshkumar Polani [aut, cre] (ORCID: <https://orcid.org/0000-0002-0418-7592>),

Salin K Eliyas [aut] (ORCID: <https://orcid.org/0000-0002-8020-5860>),
Manikandanesan Sakthivel [aut] (ORCID:
<https://orcid.org/0000-0002-5438-3970>),
Yuvaraj Krishnamoorthy [aut] (ORCID:
<https://orcid.org/0000-0003-4688-510X>),
Marie Gilbert Majella [aut] (ORCID:
<https://orcid.org/0000-0003-4036-5162>)

Maintainer Rubeshkumar Polani <rubesh@thinkdenominator.com>

Repository CRAN

Date/Publication 2025-08-18 14:50:13 UTC

Contents

check_collinearity *Check Collinearity Using VIF for Fitted Models*

Description

Computes Variance Inflation Factors (VIF) for fitted models returned by uni_reg(), multi_reg(), uni_reg_nbin(), or multi_reg_nbin(). Returns one VIF table per model. For multivariate models only

Usage

```
check_collinearity(model)
```

Arguments

model	A fitted model object with class "uni_reg", "multi_reg", "uni_reg_nbin", or "multi_reg_nbin".

Value

A tibble containing VIF values and interpretation. For multivariable models, returns one tibble. For univariate models, an error is raised indicating VIF is not applicable.

Examples

```
if (requireNamespace("gtregression", quietly = TRUE) &&
  requireNamespace("mlbench", quietly = TRUE) &&
  getRversion() >= "4.1.0") {
  data(PimaIndiansDiabetes2, package = "mlbench")
  pima <- PimaIndiansDiabetes2 |> dplyr::filter(!is.na(diabetes))
  pima$diabetes <- ifelse(pima$diabetes == "pos", 1, 0)
  fit <- multi_reg(pima,
    outcome = "diabetes",
    exposures = c("age", "mass", "glucose"),
    approach = "logit"
  )
  check_collinearity(fit)
}
```

check_convergence *Check Convergence for a Regression Model*

Description

Assesses model convergence and provides diagnostics for each exposure (in univariate mode) or for the full model (in multivariable mode), depending on the regression approach used.

Usage

```
check_convergence(
  data,
  exposures,
  outcome,
  approach = "logit",
  multivariate = FALSE
)
```

Arguments

data	A data frame containing the dataset.
exposures	A character vector of predictor variable names. If multivariate = FALSE, each exposure is assessed separately. If multivariate = TRUE, exposures are included together.
outcome	A character string specifying the outcome variable.
approach	A character string specifying the regression approach. One of: "logit", "log-binomial", "poisson", "robpoisson", or "negbin".
multivariate	Logical. If TRUE, checks convergence for a multivariable model; otherwise, performs checks for each univariate model.

Details

For robpoisson, predicted probabilities (fitted values) may exceed 1, which is acceptable when estimating risk ratios but should not be interpreted as actual probabilities.

This function is useful for identifying convergence issues, especially for "log-binomial" models, which often fail to converge .

Value

A data frame summarizing convergence diagnostics, including:

Exposure Name of the exposure variable.

Model The regression approach used.

Converged TRUE if the model converged successfully; FALSE otherwise.

Max.prob Maximum predicted probability or fitted value in the dataset.

See Also

[identify_confounder()], [interaction_models()]

Examples

```
if (requireNamespace("gtregression", quietly = TRUE)) {
  data(data_PimaIndiansDiabetes, package = "gtregression")

  check_convergence(
```

```
    data = data_PimaIndiansDiabetes,
    exposures = c("age", "bmi"),
    outcome = "diabetes",
    approach = "logit"
  )

  check_convergence(
    data = data_PimaIndiansDiabetes,
    exposures = c("age", "bmi"),
    outcome = "diabetes",
    approach = "logit",
    multivariate = TRUE
  )
}
```

`data_birthwt` *Birth Weight Data*

Description

A dataset from the **MASS** package containing risk factors associated with low birth weight (LBW) in newborns. Originally collected at Baystate Medical Center, Springfield, Massachusetts, USA.

Usage

```
data_birthwt
```

Format

A data frame with 189 observations and 10 variables:

low Indicator for birth weight < 2500g (binary): `0` = `normal`, `1` = `low birth weight`

age Mother's age in years (numeric)

lwt Mother's weight in pounds at last menstrual period (numeric)

race Mother's race (factor): `1` = `White`, `2` = `Black`, `3` = `Other`

smoke Smoking status during pregnancy (binary): `0` = `No`, `1` = `Yes`

ptl Number of previous premature labors (integer)

ht History of hypertension (binary): `0` = `No`, `1` = `Yes`

ui Presence of uterine irritability (binary): `0` = `No`, `1` = `Yes`

ftv no of physician visits during the 1st trimester (integer, 0–6)

bwt Birth weight in grams (numeric)

Details

The outcome variable is binary ('low'): birth weight < 2500g (yes = 1) or not (no = 0).

Source

Hosmer, D.W., Lemeshow, S. (1989). *Applied Logistic Regression.* New York: Wiley. Also available in **MASS** and described in detail in its documentation.

`data_epilepsy` *Epilepsy Treatment and Seizure Counts*

Description

RCT on the effect of a drug on the seizures in patients with epilepsy. Contains repeated measures data with treatment groups, baseline seizure counts, and follow-up counts.

Usage

```
data_epilepsy
```

Format

A data frame with 236 observations and 9 variables:

y Number of seizures in a 2-week period (count)

trt Treatment group (factor): `placebo` or `progabide`

base Seizure count during baseline period (numeric)

age Age of patient (numeric)

V4 Indicator for 4th visit (binary)

subject Patient ID (factor)

period Follow-up period number (integer)

lbase Log of baseline seizures (numeric)

lage Log of age (numeric)

Source

MASS package. Original data from Thall and Vail (1990)

data_gt_quin *Student Absenteeism in Rural Schools*

Description

This dataset contains observations on the number of days absent from school for children in rural Australia, along with student characteristics. It's commonly used to demonstrate count models such as Poisson and Negative Binomial regression.

Usage

```
data_gt_quin
```

Format

A data frame with 146 observations and 5 variables:

Eth Ethnicity (`"A"` = Aboriginal, `"N"` = Non-Aboriginal)

Sex Sex (`"F"` or `"M"`)

Age Age group (`"F0"`, `"F1"`, `"F2"`, `"F3"`)

Lrn Learner status (`"AL"` = average learner, `"SL"` = slow learner)

Days Number of days absent from school (count outcome)

Source

MASS package. See also Venables and Ripley (2002), *Modern Applied Statistics with S*.

data_infertility *Infertility Matched Case-Control Study*

Description

investigating the relationship between infertility and abortions.

Usage

```
data_infertility
```

Format

A data frame with 248 observations and 8 variables:

education Education level (0 = 0–5 years, 1 = 6–11 years, 2 = 12+ years)
age Age in years
parity Number of prior pregnancies
induced Number of induced abortions
case Infertility case status (1 = case, 0 = control)
spontaneous Number of spontaneous abortions
stratum Matched set ID
pooled.stratum Pooled stratum ID used for conditional regression

Source

https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/infert.html

---

data_lungcancer *Lung Cancer Trial Data*

Description

Survival data from a clinical trial of lung cancer patients conducted by the Veteran's Administration.

Usage

```
data_lungcancer
```

Format

A data frame with 137 observations and 8 variables:

trt Treatment group (1 = standard, 2 = test)
celltype Cell type (squamous, smallcell, adeno, large)
time Survival time (in days)
status Censoring status (1 = died, 0 = censored)
karno Karnofsky performance score (higher = better)
diagtime Months from diagnosis to randomization
age Age in years
prior Prior therapy (0 = no, 10 = yes)

Source

https://CRAN.R-project.org/package=survival

References

Kalbfleisch JD and Prentice RL (1980). The Statistical Analysis of Failure Time Data.

`data_PimaIndiansDiabetes` *PimaIndians2 Diabetes Dataset*

Description

A cleaned version of the original Pima Indians Diabetes dataset from the ‘mlbench‘ package. Useful for demonstrating regression approaches for binary outcomes.

Usage

```
data_PimaIndiansDiabetes
```

Format

A data frame with 768 observations and 9 variables:

pregnant Number of times pregnant

glucose Plasma glucose concentration (glucose tolerance test)

pressure Diastolic blood pressure (mm Hg)

triceps Triceps skin fold thickness (mm)

insulin 2-Hour serum insulin (mu U/ml)

mass Body mass index (BMI)

pedigree Diabetes pedigree function

age Age in years

diabetes Factor indicating diabetes status (pos/neg)

Source

https://www.openml.org/d/37

`descriptive_table` *Descriptive Summary Table for Study Characteristics (User-Friendly)*

Description

Creates a clean, publication-ready summary table using ‘gtsummary::tbl_summary()‘. Designed for beginner analysts, this function applies sensible defaults and flexible options to display categorical and continuous variables with or without stratification. It supports one-line summaries of dichotomous variables, handles missing data gracefully, and includes an optional "Overall" column for comparison.

Usage

```
descriptive_table(
  data,
  exposures,
  by = NULL,
  percent = c("column", "row"),
  digits = 1,
  show_missing = c("ifany", "no"),
  show_dichotomous = c("all_levels", "single_row"),
  show_overall = c("no", "first", "last"),
  statistic = NULL,
  value = NULL
)
```

Arguments

data	A data frame containing your study dataset.
exposures	A character vector specifying the variable names (columns) in ‘data‘ that should be included in the summary table. These can be categorical or continuous.
by	Optional. A single character string giving the name of a grouping variable (e.g., outcome). If supplied, the table will show stratified summaries by this variable.
percent	Character. Either ‘"column"‘ (default) or ‘"row"‘. - ‘"column"‘ calculates percentages within each group defined by ‘by‘ (i.e., denominator = column total). - ‘"row"‘ calculates percentages across ‘by‘ groups (i.e., denominator = row total). If ‘by‘ is not specified, ‘"column"‘ is used and ‘"row"‘ is ignored.
digits	Integer. Controls how many decimal places are shown for percentages and means. Defaults to 1.
show_missing	Character. One of ‘"ifany"‘ (default) or ‘"no"‘. - ‘"ifany"‘ shows missing value counts only when missing values exist. - ‘"no"‘ hides missing counts entirely.
show_dichotomous	Character. One of ‘"all_levels"‘ (default) or ‘"single_row"‘. - ‘"all_levels"‘ displays all levels of binary (dichotomous) variables. - ‘"single_row"‘ shows only one row (typically "Yes", "Present", or a user-defined level), making the table more compact.
show_overall	Character. One of ‘"no"‘ (default), ‘"first"‘, or ‘"last"‘. If ‘by‘ is supplied: - ‘"first"‘ includes a column for overall summaries before the stratified columns. - ‘"last"‘ includes the overall column at the end. - ‘"no"‘ disables the overall column.
statistic	Optional named vector of summary types for specific variables. For example, use ‘statistic = c(age = "mean", bmi = "median")‘ to override default summaries. Accepted values: ‘"mean"‘, ‘"median"‘, ‘"mode"‘, ‘"count"‘.
value	Optional. A list of formulas specifying which level of a binary variable to show when ‘show_dichotomous = "single_row"‘. For example, ‘value = list(sex ~ "Female")‘ will report only the "Female" row.

Value

A ‘gtsummary::tbl_summary‘ object with additional class ‘"descriptive_table"‘. Can be printed, customized, merged, or exported.

Examples

```
if (requireNamespace("mlbench", quietly = TRUE)) {
  data("PimaIndiansDiabetes2", package = "mlbench")
  library(dplyr)
pima <- PimaIndiansDiabetes2 |>
  mutate(
    diabetes = ifelse(diabetes == "pos", 1, 0),

    bmi = cut(
      mass,
      breaks = c(-Inf, 18.5, 24.9, 29.9, Inf),
      labels = c("Underweight", "Normal", "Overweight", "Obese")
    )
  )
  descriptive_table(pima, exposures = c("age", "bmi"),
                    by = "diabetes")
}
```

`dissect` *Dissect a Dataset Before Regression*

Description

Returns a tidy summary of each variable’s structure, missingness, uniqueness, and suitability for use in regression models.

Usage

```
dissect(data)
```

Arguments

`data`	A data frame.

Value

A tibble with columns: Variable, Type, Missing (and Regression Hint.

Examples

```
dissect(data_birthwt)
```

`identify_confounder` *Identify Confounders Using the Change-in-Estimate Method*

Description

Identifies whether one or more variables are confounders by comparing the crude and adjusted effect estimates of a primary exposure on an outcome. A variable is flagged as a confounder if its inclusion changes the estimate by more than a specified threshold (default = 10

Usage

```
identify_confounder(
  data,
  outcome,
  exposure,
  potential_confounder,
  approach = "logit",
  threshold = 10
)
```

Arguments

`data`	A data frame containing the variables.
`outcome`	The name of the outcome variable (character string).
`exposure`	The primary exposure variable (character string).
`potential_confounder`	One or more variables to test as potential confounders.
`approach`	The regression modeling approach. One of: `"logit"`, `"log-binomial"`, `"poisson"`, `"negbin"`, `"robpoisson"`, or `"linear"`.
`threshold`	Numeric. Percent change threshold to define confounding (default = 10). If the absolute percent change exceeds this, the variable is flagged as a confounder.

Details

Supports logistic, log-binomial, Poisson, robust Poisson, negative binomial, and linear regression approaches.

This method does not evaluate effect modification. Use causal diagrams (e.g., DAGs) and subject-matter knowledge to supplement decisions.

Value

If one confounder is provided, prints crude and adjusted estimates with a confounding flag. If multiple are given, returns a tibble with:

covariate Name of potential confounder.

crude_est Crude effect estimate.

adjusted_est Adjusted estimate including the confounder.

pct_change Percent change from crude to adjusted.

is_confounder Logical: whether confounding threshold is exceeded.

See Also

[check_convergence()], [interaction_models()]

Examples

```
data <- data_PimaIndiansDiabetes
identify_confounder(
  data = data,
  outcome = "glucose",
  exposure = "insulin",
  potential_confounder = "age_cat",
  approach = "linear"
)
```

---

interaction_models *Compare Models With and Without Interaction Term*

Description

This function fits two models—one with and one without an interaction term between an exposure and a potential effect modifier— and compares them using either a likelihood ratio test (LRT) or Wald test. It is useful for assessing whether there is statistical evidence of interaction (effect modification).

Usage

```
interaction_models(
  data,
  outcome,
  exposure,
  covariates = NULL,
  effect_modifier,
  approach = "logit",
  test = c("LRT", "Wald"),
  verbose = TRUE
)
```

Arguments

data	A data frame containing all required variables.
outcome	The name of the outcome variable
exposure	The name of the main exposure variable.
covariates	character vector of additional covariates to adjust for
effect_modifier	The name of the variable to test for interaction
approach	The regression modeling approach to use. One of: "logit", "log-binomial", "poisson", "robpoisson", "negbin", or "linear".
test	Type of statistical test for model comparison. Either: "LRT" (likelihood ratio test, default) or "Wald".
verbose	Logical; if TRUE, prints a basic interpretation of whether interaction is likely present (default = FALSE).

Value

A list with the following elements:

- model_no_interaction: The model without the interaction term.
- model_with_interaction: The model with the interaction term.
- p_value: The p-value for interaction (based on selected test).
- interpretation: A brief text interpretation if verbose = TRUE.

Examples

```
data <- data_PimaIndiansDiabetes
```

---

merge_tables *Merge Multiple gtsummary Tables (Descriptive, Univariate, Multivariable)*

Description

Flexibly merges any 2 or more ‘gtsummary‘ tables (e.g., from ‘descriptive_table()‘, ‘uni_reg()‘, ‘multi_reg()‘) into a single table using ‘tbl_merge()‘. Automatically applies column spanners based on the order of inputs.

Usage

```
merge_tables(..., spanners = NULL)
```

Arguments

...	Two or more ‘gtsummary‘ table objects to merge.
spanners	Optional character vector of column header titles. If not supplied, defaults to ‘"Table 1"‘, ‘"Univariate"‘, ‘"Multivariable"‘ etc.

Value

A merged ‘gtsummary::tbl_merge‘ object.

Examples

```
if (requireNamespace("mlbench", quietly = TRUE)) {
  data("PimaIndiansDiabetes2", package = "mlbench")
  library(dplyr)
  library(gtregression)

  # Prepare data
  pima <- PimaIndiansDiabetes2 |>
    mutate(
      diabetes = ifelse(diabetes == "pos", 1, 0),
      bmi_cat = cut(
        mass,
        breaks = c(-Inf, 18.5, 24.9, 29.9, Inf),
        labels = c("Underweight", "Normal", "Overweight", "Obese")
      )
    )

  # Descriptive table
  desc_tbl <- descriptive_table(pima,
                                exposures = c("age", "bmi_cat"),
                                by = "diabetes")

  # Univariate logistic regression
  uni_tbl <- uni_reg(
    data = pima,
    outcome = "diabetes",
    exposures = c("age", "bmi_cat"),
    approach = "logit"
  )

  # Multivariable logistic regression
  multi_tbl <- multi_reg(
    data = pima,
    outcome = "diabetes",
    exposures = c("age", "bmi_cat"),
    approach = "logit"
  )

  # Merge descriptive + univariate + multivariate
  merge_tables(desc_tbl, uni_tbl, multi_tbl)
```

```
  # Merge with custom spanners
  merge_tables(desc_tbl, uni_tbl, spanners = c("Summary", "Crude"))

  # Merge just uni and multi
  merge_tables(uni_tbl, multi_tbl)
}
```

`modify_table` *Modify Regression Table Labels and Layout*

Description

Allows customization of labels, headers, and layout of regression tables created using ‘gtsummary‘. Designed for tables from functions like ‘uni_reg()‘, ‘multi_reg()‘, etc.

Usage

```
modify_table(
  gt_table,
  variable_labels = NULL,
  level_labels = NULL,
  header_labels = NULL,
  caption = NULL,
  bold_labels = FALSE,
  bold_levels = FALSE,
  remove_N = FALSE,
  remove_N_obs = FALSE,
  remove_abbreviations = FALSE,
  caveat = NULL
)
```

Arguments

`gt_table`	A ‘gtsummary‘ table object.
`variable_labels`	A named vector for relabeling variable names.
`level_labels`	A named list for relabeling levels of variables. Should be structured as ‘list(var1 = c(old1 = new1, old2 = new2), ...)‘.
`header_labels`	A named vector for relabeling column headers. Names should match internal column names (e.g., ‘"estimate"‘, ‘"p.value"‘).
`caption`	A character string used to set the table title.
`bold_labels`	Logical. If ‘TRUE‘, bolds variable labels.
`bold_levels`	Logical. If ‘TRUE‘, bolds factor level labels.
`remove_N`	Logical. If ‘TRUE‘, hides the ‘N‘ column in univariate regression tables (‘uni_reg‘, ‘uni_reg_nbin‘). Ignored for other tables.

remove_N_obs Logical. If 'TRUE', removes the source note showing the no of observations in multivariable models ('multi_reg', 'multi_reg_nbin').

remove_abbreviations
Logical. If 'TRUE', removes default footnotes for estimate abbreviations.

caveat A character string to add as a footnote (source note) below the table, e.g., "N may vary due to missing data."

Value

A customized 'gtsummary' table object with modified labels, layout, and options.

Examples

```
if (requireNamespace("mlbench", quietly = TRUE)) {
  data("PimaIndiansDiabetes2", package = "mlbench")
  library(dplyr)
  library(gtregression)

  # Prepare data
  pima <- PimaIndiansDiabetes2 |>
    mutate(
      diabetes = ifelse(diabetes == "pos", 1, 0),
      bmi_cat = cut(
        mass,
        breaks = c(-Inf, 18.5, 24.9, 29.9, Inf),
        labels = c("Underweight", "Normal", "Overweight", "Obese")
      )
    )

  # Descriptive table
  desc_tbl <- descriptive_table(pima,
                                exposures = c("age", "bmi_cat"),
                                by = "diabetes")

  # Univariate logistic regression
  uni_rr <- uni_reg(
    data = pima,
    outcome = "diabetes",
    exposures = c("age", "bmi_cat"),
    approach = "logit"
  )
}
```

---

multi_reg *Multivariable Regression (Adjusted Odds, Risk, or Rate Ratios)*

Description

Fits multivariable regression models for binary, count, or continuous outcomes and returns a publication-ready summary table using ‘gtsummary‘. Depending on the specified ‘approach‘, the function estimates adjusted Odds Ratios (OR), Risk Ratios (RR), Incidence Rate Ratios (IRR), or Beta coefficients.

Usage

```
multi_reg(data, outcome, exposures, approach = "logit")
```

Arguments

`data`	A data frame containing the analysis variables.
`outcome`	The name of the outcome variable. Must be a character string.
`exposures`	A character vector of predictor variables to include.
`approach`	Modeling approach to use. One of: - ‘"logit"‘ for logistic regression (OR), - ‘"log-binomial"‘ for log-binomial regression (RR), - ‘"poisson"‘ for Poisson regression (IRR), - ‘"robpoisson"‘ for robust Poisson regression (RR), - ‘"linear"‘ for linear regression (Beta coefficients), - ‘"negbin"‘ for negative binomial regression (IRR).

Value

An object of class ‘multi_reg‘, extending ‘gtsummary::tbl_regression‘. Additional components can be accessed using:

- `x$models`: List of fitted model objects.
- `x$model_summaries`: List of summary outputs.
- `x$reg_check`: Regression diagnostics (only for linear models).
- `x$table`: Returns the main regression table.

Accessors

`$models` List of fitted model objects.

`$model_summaries` A tibble of tidy regression summaries for each model.

See Also

[uni_reg()], [plot_reg()], [plot_reg_combine()]

Examples

```
if (requireNamespace("mlbench", quietly = TRUE)) {
  data(PimaIndiansDiabetes2, package = "mlbench")
  pima <- dplyr::mutate(PimaIndiansDiabetes2,
  diabetes = ifelse(diabetes == "pos", 1, 0))
  model <- multi_reg(
    data = pima,
```

```
    outcome = "diabetes",
    exposures = c("age", "mass"),
    approach = "logit"
  )
 print(model)
}
```

---

plot_reg *Visualize a Regression Model as a Forest Plot*

Description

Creates a forest plot from a ‘gtsummary‘ object. Supports both univariate and multivariable models with hierarchical labels for categorical variables. Designed to work seamlessly with outputs from functions like ‘uni_reg()‘ and ‘multi_reg()‘.

Usage

```
plot_reg(
  tbl,
  title = NULL,
  ref_line = 1,
  order_y = NULL,
  log_x = FALSE,
  xlim = NULL,
  breaks = NULL,
  point_color = "#1F77B4",
  errorbar_color = "#4C4C4C",
  base_size = 14,
  show_ref = TRUE
)
```

Arguments

tbl	A ‘gtsummary‘ object from regression functions
title	Optional plot title (character).
ref_line	Numeric value for the reference line (default = 1).
order_y	Optional character vector to the customise y-axis order
log_x	Logical. If ‘TRUE‘, uses a logarithmic x-axis (default = FALSE).
xlim	Optional numeric vector specifying x-axis limits
breaks	Optional numeric vector for x-axis tick breaks.
point_color	Color of the points (default is automatic).
errorbar_color	Color of the error bars (default is automatic).
base_size	Base font size for text elements.
show_ref	Logical. If ‘TRUE‘, includes reference in the plot.

Value

A ‘ggplot2‘ object representing the forest plot.

Examples

```
if (requireNamespace("mlbench", quietly = TRUE)) {
  data("PimaIndiansDiabetes2", package = "mlbench")
  library(dplyr)
  library(gtregression)

  # Prepare data
  pima <- PimaIndiansDiabetes2 |>
    mutate(
      diabetes = ifelse(diabetes == "pos", 1, 0),
      bmi_cat = cut(
        mass,
        breaks = c(-Inf, 18.5, 24.9, 29.9, Inf),
        labels = c("Underweight", "Normal", "Overweight", "Obese"))
  )

  # Univariate logistic regression
  uni_rr <- uni_reg(
    data = pima,
    outcome = "diabetes",
    exposures = c("age", "bmi_cat"),
    approach = "logit"
  )
plot_reg(uni_rr)
}
```

---

plot_reg_combine *Visualize Univariate and Multivariate Regression Side-by-Side*

Description

Generates side-by-side plots to compare univariate & multivariable results

Usage

```
plot_reg_combine(
  tbl_uni,
  tbl_multi,
  title_uni = NULL,
  title_multi = NULL,
  ref_line = 1,
  order_y = NULL,
  log_x = FALSE,
  point_color = "#1F77B4",
```

```
  errorbar_color = "#4C4C4C",
  base_size = 14,
  show_ref = TRUE,
  xlim_uni = NULL,
  breaks_uni = NULL,
  xlim_multi = NULL,
  breaks_multi = NULL
)
```

Arguments

tbl_uni	A 'gtsummary' object from 'uni_reg()' etc.,
tbl_multi	A 'gtsummary' object from 'multi_reg()'.
title_uni	Optional plot title for the univariate model
title_multi	Optional plot title for the multivariable mode
ref_line	Numeric value for the reference line (default = 1).
order_y	Optional character vector to manually order the y-axis labels.
log_x	Logical. If 'TRUE', x-axis is log-transformed (default = FALSE).
point_color	Optional color for plot points.
errorbar_color	Optional color for error bars.
base_size	Numeric. Base font size for plot text elements.
show_ref	Logical. If 'TRUE', includes reference categories
xlim_uni	Optional numeric vector to set x-axis limits for uni plot.
breaks_uni	Optional numeric vector to set x-axis breaks for uni plot.
xlim_multi	Optional numeric vector to set x-axis limits for multi plot
breaks_multi	Optional numeric vector to set x-axis breaks- multi plot.

Value

A 'ggplot2' object with two forest plots displayed side-by-side.

Examples

```
if (requireNamespace("mlbench", quietly = TRUE)) {
  data("PimaIndiansDiabetes2", package = "mlbench")
  library(dplyr)
  library(gtregression)

  # Prepare data
  pima <- PimaIndiansDiabetes2 |>
    mutate(
      diabetes = ifelse(diabetes == "pos", 1, 0),
      bmi_cat = cut(
        mass,
        breaks = c(-Inf, 18.5, 24.9, 29.9, Inf),
        labels = c("Underweight", "Normal", "Overweight", "Obese")
```

```
    ),
    age_cat = cut(
      age,
      breaks = c(-Inf, 29, 49, Inf),
      labels = c("Young", "Middle-aged", "Older")
    )
  )

# Univariate logistic regression
uni_rr <- uni_reg(
  data = pima,
  outcome = "diabetes",
  exposures = c("age_cat", "bmi_cat"),
  approach = "logit"
)

# Multivariable logistic regression
multi_rr <- multi_reg(
  data = pima,
  outcome = "diabetes",
  exposures = c("age_cat", "bmi_cat"),
  approach = "logit"
)

# Combined plot
plot_reg_combine(uni_rr, multi_rr)
}
```

---

save_docx *Save Multiple Tables and Plots to a Word Document*

Description

Saves a collection of gtsummary tables and ggplot2 plots into a .docx file.

Usage

```
save_docx(tables = NULL, plots = NULL, filename = "report.docx", titles = NULL)
```

Arguments

tables	A list of gtsummary tables.
plots	A list of ggplot2 plot objects.
filename	File name for the output (with or without .docx extension).
titles	Optional. A character vector of titles.

Value

A Word document saved to a temporary directory (if no path is given). No object is returned.

Examples

```
library(gtsummary)
library(ggplot2)
tbl <- tbl_regression(glm(mpg ~ hp + wt, data = mtcars))
p <- ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()
save_docx(
  tables = list(tbl),
  plots = list(p),
  filename = file.path(tempdir(), "report.docx"),
  titles = c("Table 1: Regression", "Figure 1: Scatterplot")
)
```

---

save_plot *Save a Single Plot*

Description

Saves a ggplot2 plot to a file in PNG, PDF, or JPG format.

Usage

```
save_plot(
  plot,
  filename = "plot",
  format = c("png", "pdf", "jpg"),
  width = 8,
  height = 6,
  dpi = 300
)
```

Arguments

plot	A ggplot2 object.
filename	Name of the file to save, with or without extension.
format	Output format. One of "png", "pdf", or "jpg".
width	Width of the saved plot in inches.
height	Height of the saved plot in inches.
dpi	Resolution of the plot in dots per inch (default is 300).

Value

Saves the file to a temporary directory (if no path is given).

Examples

```
library(ggplot2)
p <- ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()
save_plot(p, filename = file.path(tempdir(), "scatterplot"), format = "png")
```

`save_table` *Save a Single Regression Table*

Description

Saves a gtsummary table as a Word, PDF, or HTML file

Usage

```
save_table(tbl, filename = "table", format = c("docx", "pdf", "html"))
```

Arguments

`tbl`	A gtsummary object (e.g., tbl_regression(), tbl_summary()).
`filename`	File name to save the output. Extension is optional.
`format`	Output format. One of "docx", "pdf", or "html".

Value

Saves the file to a temporary directory (if no path is given). Does not return an object.

Examples

```
model <- glm(mpg ~ hp + wt, data = mtcars)
tbl <- gtsummary::tbl_regression(model)
save_table(tbl, filename = file.path(tempdir(), "regression_table"), format = "docx")
```

`select_models` *Stepwise Model Selection with Evaluation Metrics*

Description

Performs stepwise model selection using forward, backward, or both directions across different regression approaches. Returns a summary table with evaluation metrics (AIC, BIC, log-likelihood, deviance) and the best model.

Usage

```
select_models(
  data,
  outcome,
  exposures,
  approach = "logit",
  direction = "forward"
)
```

Arguments

`data`	A data frame containing the outcome and predictor variables.
`outcome`	A character string indicating the outcome variable.
`exposures`	vector of predictor variables to consider in the model.
`approach`	Regression method. One of: `"logit"`, `"log-binomial"`, `"poisson"`, `"robpoisson"`, `"negbin"`, or `"linear"`.
`direction`	Stepwise selection direction. One of: `"forward"` (default), `"backward"`, or `"both"`.

Value

A list with the following components:

- `results_table`: A tibble summarising each tested model's metric (AIC, BIC, deviance, log-likelihood, adjusted R^2 if applicable).
- `best_model`: The best-fitting model object based on low AIC.
- `all_models`: A named list of all fitted models.

Examples

```
data <- data_PimaIndiansDiabetes
stepwise <- select_models(
  data = data,
  outcome = "glucose",
  exposures = c("age", "pregnant", "mass"),
  approach = "linear",
```

```
  direction = "forward"
)
summary(stepwise)
stepwise$results_table
stepwise$best_model
```

`stratified_multi_reg` *Stratified Multivariable Regression (Adjusted OR, RR, IRR, or Beta)*

Description

Performs multivariable regression with multiple exposures on a binary, count, or continuous outcome, stratified by a specified variable. NA values in the stratifier are excluded from analysis.

Usage

```
stratified_multi_reg(data, outcome, exposures, stratifier, approach = "logit")
```

Arguments

`data`	A data frame containing the variables.
`outcome`	name of the outcome variable.
`exposures`	vector specifying the predictor (exposure) variables.
`stratifier`	A character string specifying the stratifying variable.
`approach`	Modeling approach to use. One of: ‘"logit"‘ (Adjusted Odds Ratios), ‘"log-binomial"‘ (Adjusted Risk Ratios), ‘"poisson"‘ (Adjusted IRRs), ‘"robpoisson"‘ (Adjusted RRs), or ‘"linear"‘ (Beta coefficients), ‘"negbin"‘ (Adjusted IRRs).

Value

An object of class ‘stratified_multi_reg‘, which includes: - ‘table‘: A ‘gtsummary::tbl_stack‘ object of regression tables by stratum, - ‘models‘: A named list of model objects for each stratum, - ‘model_summaries‘: A list of tidy model summaries, - ‘reg_check‘: Diagnostics results (if available for the model type).

Accessors

`$table` Stacked table of stratified regression outputs.

`$models` Named list of fitted models per stratum.

`$model_summaries` Tidy summaries for each model.

`$reg_check` Regression diagnostic checks (when applicable).

See Also

[multi_reg()], [stratified_uni_reg()], [plot_reg()]

Examples

```
if (requireNamespace("mlbench", quietly = TRUE) &&
  requireNamespace("dplyr", quietly = TRUE)) {
  data(PimaIndiansDiabetes2, package = "mlbench")
  pima <- dplyr::mutate(
    PimaIndiansDiabetes2,
    diabetes = ifelse(diabetes == "pos", 1, 0),
    glucose_cat = dplyr::case_when(
      glucose < 140 ~ "Normal",
      glucose >= 140 ~ "High"
    )
  )
  stratified_multi <- stratified_multi_reg(
    data = pima,
    outcome = "diabetes",
    exposures = c("age", "mass"),
    stratifier = "glucose_cat",
    approach = "logit"
  )
  stratified_multi$table
}
```

`stratified_uni_reg` *Performs univariate regression for each exposure on a binary, count, or continuous outcome, stratified by a specified variable. Produces a stacked 'gtsummary' table with one column per stratum, along with underlying models and diagnostics.*

Description

Performs univariate regression for each exposure on a binary, count, or continuous outcome, stratified by a specified variable. Produces a stacked 'gtsummary' table with one column per stratum, along with underlying models and diagnostics.

Usage

```
stratified_uni_reg(data, outcome, exposures, stratifier, approach = "logit")
```

Arguments

`data`	A data frame containing the variables.
`outcome`	name of the outcome variable.
`exposures`	A vector specifying the predictor (exposure) variables.
`stratifier`	A character string specifying the stratifier
`approach`	Modeling approach to use. One of: '"logit"' (Odds Ratios), '"log-binomial"' (Risk Ratios), '"poisson"' (Incidence Rate Ratios), '"robpoisson"' (Robust RR), '"linear"' (Beta coefficients), '"negbin"' (Incidence Rate Ratios),.

Value

An object of class ‘stratified_uni_reg‘, which includes: - ‘table‘: A ‘gtsummary::tbl_stack‘ object with stratified results, - ‘models‘: A list of fitted models for each stratum, - ‘model_summaries‘: A tidy list of model summaries, - ‘reg_check‘: A tibble of regression diagnostics (when available).

Accessors

`$table` Stacked stratified regression table.

`$models` List of fitted model objects for each stratum.

`$model_summaries` List of tidy model summaries.

`$reg_check` Diagnostic check results (when applicable).

See Also

[multi_reg()], [plot_reg()], [identify_confounder()]

Examples

```
if (requireNamespace("mlbench", quietly = TRUE) &&
  requireNamespace("dplyr", quietly = TRUE)) {
  data(PimaIndiansDiabetes2, package = "mlbench")
  pima <- dplyr::mutate(
    PimaIndiansDiabetes2,
    diabetes = ifelse(diabetes == "pos", 1, 0),
    glucose_cat = dplyr::case_when(
      glucose < 140 ~ "Normal",
      glucose >= 140 ~ "High"
    )
  )
  stratified_uni <- stratified_uni_reg(
    data = pima,
    outcome = "diabetes",
    exposures = c("age", "mass"),
    stratifier = "glucose_cat",
    approach = "logit"
  )
  stratified_uni$table
}
```

---

uni_reg *Univariate regression (Odds, Risk, or Rate Ratios)*

Description

Performs univariate regression for each exposure on a binary, continuous, or count outcome. Depending on ‘approach‘, returns either Odds Ratios (OR), Risk Ratios (RR), or Incidence Rate Ratios (IRR).

Usage

```
uni_reg(data, outcome, exposures, approach = "logit")
```

Arguments

data	A data frame containing the variables.
outcome	outcome variable (binary, continuous, or count).
exposures	A vector of predictor variables.
approach	Modeling approach to use. One of: '"logit"' (OR), '"log-binomial"' (RR), '"poisson"' (IRR), '"robpoisson"' (RR), '"linear"' (Beta coefficients), '"neg-bin"' (IRR)

Details

This function requires the following packages: 'dplyr', 'purrr', 'gtsummary', 'risks'.

Value

A list of class 'uni_reg' and 'gtsummary::tbl_stack', including:

- A publication-ready regression table ('tbl_stack')
- Accessor elements:
  - '$models': Fitted regression models for each exposure
  - '$model_summaries': Tidy model summaries
  - '$reg_check': Diagnostics (only for linear regression)

See Also

multi_reg, plot_reg

Examples

```
data(PimaIndiansDiabetes2, package = "mlbench")
library(dplyr)
pima <- PimaIndiansDiabetes2 |>
  dplyr::mutate(diabetes = ifelse(diabetes == "pos", 1, 0))
uni_reg(pima, outcome = "diabetes", exposures = "age", approach = "logit")
```

Index