

Package ‘gtregression’

August 21, 2026

Type Package

Title Tools for Creating Publication-Ready Regression Tables

Version 1.1.0

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](https://doi.org/10.1093/aje/kwi188)>), Firth penalized logistic regression (Firth 1993 <[doi:10.1093/biomet/80.1.27](https://doi.org/10.1093/biomet/80.1.27)>), Poisson and robust Poisson regression (Zou 2004 <[doi:10.1093/aje/kwh090](https://doi.org/10.1093/aje/kwh090)>), negative binomial regression (Hilbe 2011, ISBN:9780521179515), Cox proportional hazards regression, parametric survival regression, causal mediation analysis, 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

Language en-US

LazyData true

Depends R (>= 4.1.0)

Imports dplyr, risks, purrr, MASS, rlang, stats, lmtest, patchwork, ggplot2, utils, sandwich, tibble, broom, broom.helpers, gt, officer, flextable, scales (>= 1.3.0), forestploter, logistf, survival

VignetteBuilder knitr

Suggests haven, mlbench (>= 2.1-11), testthat (>= 3.1.5), knitr, rmarkdown, car, forcats, pkgdown, ggtext, shiny, rstudioapi, DT, readxl, rio

Config/testthat/edition 3

Config/Needs/website pkgdown

URL <https://gtregression.thinkdenominator.com/>,

<https://github.com/ThinkDenominator/gtregression>

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

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Rubeshkumar Polani [aut, cre] (ORCID:

[<https://orcid.org/0000-0002-0418-7592>](https://orcid.org/0000-0002-0418-7592)),

Salin K Eliyas [aut] (ORCID: [<https://orcid.org/0000-0002-8020-5860>](https://orcid.org/0000-0002-8020-5860)),

Manikandanesan Sakthivel [aut] (ORCID:

[<https://orcid.org/0000-0002-5438-3970>](https://orcid.org/0000-0002-5438-3970)),

Mogan Kaviprawin [aut] (ORCID: [<https://orcid.org/0000-0002-1659-059X>](https://orcid.org/0000-0002-1659-059X)),

Yuvaraj Krishnamoorthy [aut] (ORCID:

[<https://orcid.org/0000-0003-4688-510X>](https://orcid.org/0000-0003-4688-510X)),

Marie Gilbert Majella [aut] (ORCID:

[<https://orcid.org/0000-0003-4036-5162>](https://orcid.org/0000-0003-4036-5162))

Maintainer Rubeshkumar Polani <rubesh@thinkdenominator.com>

Repository CRAN

Date/Publication 2026-08-21 07:30:02 UTC

Contents

check_collinearity	3
check_convergence	4
check_ph	6
compare_models	7
cox_reg	9
data_birthwt	12
data_diabetes_mediation	13
data_endometrial	14
data_epilepsy	15
data_gt_quin	16
data_infertility	16
data_lungcancer	17
data_SynthDiabetes	18
descriptive_table	18
dissect	20
forest_df	21
forest_reg	22
gtregression_app	24
identify_confounder	25
interaction_models	28
km_plot	30
km_risk_table	32
logrank_test	34
mediation_analysis	36

merge_tables	38
modify_table	39
multi_reg	41
plot_mediation	43
plot_model_fit	44
plot_reg	45
plot_reg_combine	47
plot_surv_fit	49
print.gtregrression	51
rmst_table	51
save_docx	53
save_forest	54
save_plot	55
save_table	56
select_models	58
stratified_multi_reg	59
stratified_uni_reg	62
survival_prob	64
survival_quantiles	65
survival_summary	67
surv_model_compare	69
surv_predict	70
surv_reg	72
uni_reg	75
\$.gtregression	77

Index	79
--------------	-----------

check_collinearity	<i>Check collinearity using VIF for fitted models</i>
--------------------	---

Description

Computes Variance Inflation Factors (VIF) for fitted multivariable models returned by `multi_reg()` or related functions.

Usage

```
check_collinearity(model, format = c("flextable", "gt", "tibble"))
```

Arguments

model	A fitted model object returned by <code>multi_reg()</code> , <code>stratified_multi_reg()</code> , or compatible <code>gtregression</code> functions. Univariable model objects are rejected because VIF is not applicable.
format	Output format. One of "flextable" (default), "gt", or "tibble". Use <code>format = "tibble"</code> to preserve the original tibble or nested-list output.

Details

If the **car** package is installed, `check_collinearity()` uses `car::vif()`. Otherwise, it computes VIF/GVIF from the fitted model matrix so that diagnostics remain available without an additional dependency.

Value

For multivariable models, a tibble if a single fitted model is present, or a named list of tibbles if multiple fitted models are present. With `format = "gt"` or `format = "flextable"`, leaf tibbles are converted to formatted tables while preserving any list nesting. The tibble contains:

Variable Model term.

VIF Variance inflation factor. For multi-degree-of-freedom terms, this is the adjusted GVIF: $GVIF^{(1 / (2 * Df))}$.

Interpretation Simple interpretation based on common cut points: no collinearity, moderate, or high.

For univariate models, an error is raised indicating that VIF is not applicable.

check_convergence	<i>Check regression model convergence</i>
-------------------	---

Description

Check whether requested models converge, either one exposure at a time or as a single multivariable model.

Usage

```
check_convergence(
  data,
  exposures,
  outcome,
  approach = "logit",
  multivariate = FALSE,
  format = c("flextable", "gt", "tibble")
)
```

Arguments

data	A data frame containing the dataset.
exposures	A character vector of predictor variable names. Quoted names are recommended in scripts, and bare names are also accepted. If <code>multivariate = FALSE</code> , each exposure is assessed separately. If <code>multivariate = TRUE</code> , exposures are included together.
outcome	A character string specifying the outcome variable. Quoted and bare names are accepted.

approach	A character string specifying the regression approach. One of: "logit", "logbinomial", "poisson", "robpoisson", or "negbin".
multivariate	Logical. If TRUE, checks convergence for a multivariable model; otherwise, performs checks for each univariate model.
format	Output format. One of "flextable" (default), "gt", or "tibble". Use format = "tibble" for the original data-frame style output.

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 is especially useful for "logbinomial" models, where convergence problems are common and robust Poisson may be a practical alternative.

Value

A data frame, gt_tbl, or flextable 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_SynthDiabetes, package = "gtregression")

  check_convergence(
    data = data_SynthDiabetes,
    exposures = c("age", "mass"),
    outcome = "diabetes",
    approach = "logit",
    format = "tibble"
  )

  check_convergence(
    data = data_SynthDiabetes,
    exposures = c("age", "mass"),
    outcome = "diabetes",
    approach = "logit",
    multivariate = TRUE,
    format = "gt"
  )
}
```

check_ph

*Check proportional hazards assumption for Cox models***Description**

Test the proportional hazards assumption for fitted Cox proportional hazards models using Schoenfeld residuals.

Usage

```
check_ph(
  model,
  transform = c("km", "rank", "identity"),
  alpha = 0.05,
  format = c("flextable", "gt", "tibble")
)
```

Arguments

model	A fitted coxph model or a cox_reg() object.
transform	Time transformation passed to survival::cox.zph(). One of "km", "rank", or "identity".
alpha	Significance level used for the simple interpretation column.
format	Output format. One of "flextable" (default), "gt", or "tibble".

Details

check_ph() is a diagnostic aid for Cox models. A small p-value suggests possible evidence against the proportional hazards assumption for a term or for the global model test. This should be interpreted with plots, clinical knowledge, follow-up patterns, and modelling purpose.

Value

A tibble, gt_tbl, or flextable with Schoenfeld residual proportional hazards tests. The table contains:

Model Model name. For a direct coxph object, this is "cox_model".

Term Model term or "GLOBAL".

Chi.square Chi-square statistic.

df Degrees of freedom.

p.value Test p-value.

Interpretation Simple screening interpretation using alpha.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                        labels = c("Standard", "Test"))

cox_fit <- cox_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c(trt, celltype, age)
)

check_ph(cox_fit)
check_ph(cox_fit, format = "tibble")
```

compare_models

Compare Prespecified Regression Models

Description

Compare gtmregression candidate models side by side using model-fit statistics. This is intended for transparent model comparison after you have already fitted the candidate models with functions such as `multi_reg()`, `cox_reg()`, or `surv_reg()`.

Usage

```
compare_models(
  ...,
  model_names = NULL,
  nested = TRUE,
  primary_exposure = NULL,
  exponentiate = NULL,
  digits = 2,
  p_digits = 3,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

<code>...</code>	Two or more gtmregression model objects, or one list containing them. Inputs should be outputs from <code>multi_reg()</code> , <code>cox_reg()</code> , or <code>surv_reg()</code> .
<code>model_names</code>	Optional character vector of names to display. If omitted, names supplied in <code>...</code> are used; otherwise models are labelled Model 1, Model 2, etc.
<code>nested</code>	Logical. If TRUE, likelihood-ratio statistics are calculated sequentially by comparing each model with the previous model. Use this only when models are nested and supplied in the intended order.

primary_exposure	Optional exposure or exact coefficient name to track across models. For Cox models this can be used to show the hazard ratio and percentage change in the log-effect estimate across candidate models.
exponentiate	Logical. If NULL, Cox, logistic, Poisson, negative-binomial, and parametric survival models are exponentiated by default, while linear models are not.
digits	Number of digits for model statistics and estimates.
p_digits	Number of digits for p-values.
format	Output format. Defaults to "flextable".
theme	Table theme preset.

Details

`compare_models()` does not refit models and does not perform hidden complete-case filtering. It compares models already fitted by `gtregression` and extracts the single fitted model stored in each object's `models` element. The reported N, event counts, and fit statistics therefore come from the model already fitted by `multi_reg()`, `cox_reg()`, or `surv_reg()`. This keeps model comparison separate from model selection: compare candidate models first, then choose the final model using clinical, epidemiological, and statistical judgement. The guided builder in `gtregression_app()` can create the fitted `gtregression` objects before calling this function; the generated app code exposes every model formula and comparison call.

Likelihood-ratio p-values are meaningful only for nested models fitted to the same analysis sample. `compare_models()` checks whether the fitted models appear to use the same analysis sample using retained model row identifiers when available; otherwise it compares N and event counts. It also checks whether sequential model pairs appear to be nested when `nested = TRUE`. Rendered warnings are context-aware: no warning about different analysis samples is shown when the compared models use the same observations, and no nested-model warning is shown when sequential models appear nested. AIC, BIC, log-likelihood, and likelihood-ratio statistics remain visible when warnings are needed, but should then be interpreted with the displayed caution.

Value

A `gtregression` object with:

- `table`: publication-ready table
- `table_body`: raw comparison statistics
- `table_display`: formatted display data
- `models`: fitted models compared
- `comparison_status`: whether models appear to use the same analysis sample
- `comparison_warnings`: caution messages that are highlighted in rendered tables when interpretation needs extra care

See Also

`gtregression_app()`, `multi_reg()`, `cox_reg()`, `surv_reg()`, `select_models()`

Examples

```
data("data_lungcancer", package = "gtregression")

lung_data <- data_lungcancer

cox_1 <- cox_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = trt
)

cox_2 <- cox_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = trt,
  adjust_for = c(age, karno)
)

compare_models(
  cox_1,
  cox_2,
  primary_exposure = trt
)

compare_models(
  cox_1,
  cox_2,
  model_names = c("Treatment only", "Treatment + age + performance"),
  primary_exposure = trt
)
```

cox_reg*Cox proportional hazards regression*

Description

Fit Cox proportional hazards models and report hazard ratios.

Usage

```
cox_reg(
  data,
  time,
  event,
  exposures,
  adjust_for = NULL,
```

```

    stratifier = NULL,
    interaction = NULL,
    multivariable = FALSE,
    multivariate = NULL,
    format = c("flectable", "gt"),
    theme = c("minimal"),
    show_sample = "events",
    model_stats = FALSE,
    show_ref = TRUE
  )

```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time, event status, and exposure variables.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>exposures</code>	Character vector of exposure variable names. Quoted names are recommended in scripts, and bare names are also accepted.
<code>adjust_for</code>	Optional character vector of adjustment variables. When supplied, one adjusted Cox model is fitted per exposure.
<code>stratifier</code>	Optional single stratifying variable. When supplied, stratum-specific Cox tables are produced using the same crude, adjusted, or multivariable workflow requested by the other arguments. The stratifier cannot also be used as the time, event, exposure, adjustment, or interaction variable.
<code>interaction</code>	Optional character scalar specifying one interaction term using standard formula syntax, e.g. <code>"trt*prior"</code> . Quoted and bare interaction syntax are accepted. In exposure-by-exposure mode, supply a single exposure; in <code>multivariable = TRUE</code> mode, the interaction is added to the single multivariable model.
<code>multivariable</code>	Logical; if <code>FALSE</code> (default), the current exposure-by-exposure workflow is used. If <code>TRUE</code> , one multivariable Cox model is fitted using all variables in exposures, and all exposure coefficients are reported.
<code>multivariate</code>	Optional logical alias for <code>multivariable</code> . This is accepted for convenience; <code>multivariable</code> is used internally.
<code>format</code>	Output table format; one of <code>"flectable"</code> (default) or <code>"gt"</code> .
<code>theme</code>	Table styling preset.
<code>show_sample</code>	For stratified Cox tables, controls which sample-size columns are shown in the publication table. One of <code>"events"</code> (default), <code>"n"</code> , <code>"both"</code> , or <code>"none"</code> . Model statistics, when requested, still retain both N and event counts.
<code>model_stats</code>	Logical; if <code>TRUE</code> , extract model-fit statistics including AIC, BIC, log-likelihood, concordance, number of events, and N.
<code>show_ref</code>	Logical; if <code>TRUE</code> (default), display reference-category rows as <code>"Ref."</code> . If <code>FALSE</code> , hide reference rows; a message reminds users to use <code>show_ref = TRUE</code> when reference rows are needed.


```
data = lung_data,
time = time,
event = status,
exposures = c("trt", "celltype", "karno", "age")
)

cox_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c(trt, celltype, prior),
  adjust_for = c(age, karno)
)

# Interaction in an adjusted exposure model
cox_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = trt,
  adjust_for = c(age, karno),
  interaction = trt*prior
)

cox_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c(trt, celltype, prior, age, karno),
  multivariable = TRUE
)

# multivariate is accepted as an alias
cox_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c(trt, age, karno),
  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_diabetes_mediation

Diabetes Mediation Teaching Dataset

Description

A health-related teaching dataset prepared from synthetic diabetes data. It is designed for the practical question: could part of the obesity and diabetes relationship operate through plasma glucose, after adjustment for basic clinical covariates?

Usage

data_diabetes_mediation

Format

A data frame with variables:

diabetes Diabetes status (No, Yes)
obesity Obesity status based on BMI ≥ 30 (No, Yes)
glucose Plasma glucose concentration
bmi Body mass index
age Age in years
blood_pressure Diastolic blood pressure
pregnancies Number of pregnancies
diabetes_pedigree Diabetes pedigree function

Details

This dataset is intended for practice and teaching. Mediation estimates from observational data should be interpreted causally only when the temporal ordering and no-unmeasured-confounding assumptions are justified by study design, DAGs, and subject-matter knowledge.

Source

Derived from data_SynthDiabetes, which is based on `mlbench::SynthDiabetes2` (**mlbench** version 2.1-11 or later).

data_endometrial	<i>Endometrial Cancer Histology Grade Data</i>
------------------	--

Description

A classic endometrial cancer dataset used to demonstrate separation in logistic regression. The outcome is high histology grade. Neovascularization is completely absent among low-grade cases in this dataset, making it useful for teaching Firth penalized logistic regression.

Usage

```
data_endometrial
```

Format

A data frame with 79 observations and 4 variables:

NV Neovascularization status (0 = absent, 1 = present)
PI Pulsatility index of the uterine artery
EH Endometrium height
HG Histology grade (0 = low grade, 1 = high grade)

Source

brglm2 package. The packaged dataset was sourced from <https://users.stat.ufl.edu/~aa/glm/data/>, the data repository used in Agresti (2015). Originally analyzed in Heinze and Schemper (2002).

References

- Agresti A (2015). *Foundations of Linear and Generalized Linear Models*. Wiley.
- Heinze G, Schemper M (2002). A solution to the problem of separation in logistic regression. *Statistics in Medicine*, 21, 2409-2419. doi:10.1002/sim.1047

data_epilepsy	<i>Epilepsy Treatment and Seizure Counts</i>
---------------	--

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	<i>Lung Cancer Trial Data</i>
-----------------	-------------------------------

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_SynthDiabetes	<i>Synthetic Diabetes Dataset</i>
--------------------	-----------------------------------

Description

A synthetic replacement for the former diabetes dataset from the **mlbench** package. It has the same structure and number of observations but does not represent real people. Useful for demonstrating regression approaches for binary outcomes.

Usage

```
data_SynthDiabetes
```

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

mlbench::SynthDiabetes2, available in **mlbench** version 2.1-11 or later.

descriptive_table	<i>Descriptive Summary Table (no gtsummary) using gt/flextable</i>
-------------------	--

Description

Publication-ready summary of categorical and continuous variables (optionally stratified). Mimics the OG gtsummary style: * column headers include N, e.g. "Overall, N=200" * categorical rows shown as n (%) * continuous rows default to Median (IQR) (footnote reflects summary)

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,
  format = c("flextable", "gt"),
  theme = c("minimal")
)
```

Arguments

data	data.frame
exposures	Character vector of variables to summarise. Quoted names are recommended in scripts, and bare names are also accepted.
by	Optional single grouping variable. Quoted and bare names are accepted.
percent	"column" (default) or "row"; aliases like "col"/"rows" accepted
digits	integer; decimals for % and continuous stats (default 1)
show_missing	"ifany" (default) or "no"
show_dichotomous	"all_levels" (default) or "single_row"
show_overall	"no" (default), "first", or "last"
statistic	Optional summary type for numeric variables. Use a single value such as "mean" or mean for all numeric variables, or a named vector such as c(age = mean, ftv = categorical). Supported values are "mean", "median", "mode", "count", and "categorical". Use "categorical" for numeric ordinal variables that should be shown as n (%).
value	optional named list for single-row binaries (e.g., list(sex="Female")); formula entries like list(sex ~ "Female") are also accepted
format	"flextable" (default) or "gt"
theme	preset or primitives

Details

If variables have a "label" attribute, for example from `labelled::var_label()`, those labels are used automatically in the displayed table. Internal matching still uses the original column names.

Value

A list with class `c("gtregression", "descriptive_table", ...)` containing:

`table` A `gt_tbl` or `flextable`.

`table_display` Display-ready data.

`table_body` Long audit data with variable, level, and type.

`variable_labels` Named character vector of display labels used for variables.

metadata Additional metadata fields.

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, verbose = FALSE, format = c("flextable", "gt", "tibble"))
```

Arguments

`data` A data frame.

`verbose` Logical; if TRUE, print the summary and interpretation notes. The tibble is returned invisibly only when printed by the console.

`format` Output format. One of "flextable" (default), "gt", or "tibble". Use format = "tibble" for pipeline-friendly raw output.

Value

A tibble, `gt_tbl`, or `flextable`, depending on format. The tibble has columns: Variable, Type, Missing (Levels, Compatibility, and Hint.

Examples

```
dissect(data_birthwt, format = "tibble")
dissect(data_birthwt, format = "gt")

# Print notes that help interpret the data structure
dissect(data_birthwt, verbose = TRUE, format = "tibble")
```

forest_df	<i>Build a compatible data frame for forest plots</i>
-----------	---

Description

Creates the tabular input used by `forest_reg()` from `uni_reg()`, `multi_reg()`, and optionally `descriptive_table()` objects. This function is useful when users want to inspect, edit, or reuse the exact data that will be passed to the forest plot. Stratified regression objects are also supported when supplied one at a time; the returned data frame keeps the characteristic rows once and places strata side by side as separate effect columns for `forest_reg()`.

Usage

```
forest_df(uni, multi = NULL, desc = NULL, digits = 2)
```

Arguments

<code>uni</code>	A <code>gtregression</code> object from <code>uni_reg()</code> . If <code>multi</code> is supplied without <code>uni</code> , the multivariable object is plotted as a single effect column.
<code>multi</code>	Optional <code>gtregression</code> object from <code>multi_reg()</code> .
<code>desc</code>	Optional descriptive table object from <code>descriptive_table()</code> .
<code>digits</code>	Number of digits used when formatting confidence intervals.

Value

A data frame with display columns, formatted effect-size columns, standard-error columns, and plotting attributes used by `forest_reg()` to draw confidence intervals.

Examples

```
birthwt_data <- data_birthwt |>
  dplyr::mutate(
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    ht = factor(ht, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW"))
  )

uni_or <- uni_reg(
  birthwt_data,
  outcome = "low",
  exposures = c("age", "lwt", "smoke", "ht"),
  approach = "logit"
)

multi_or <- multi_reg(
  birthwt_data,
  outcome = "low",
  exposures = c("smoke", "ht"),
  adjust_for = c("age", "lwt"),
```

```

    approach = "logit"
  )

forest_data <- forest_df(uni_or, multi_or)
head(forest_data)

```

forest_reg

Draw a forest table from regression outputs

Description

Wrapper around `forestploter::forest()` that works directly with `forest_df()` output or with `gtregression` regression objects. It can show descriptive columns and one or more model effect columns in a table-style forest plot while leaving axis and interval drawing to **forestploter**.

Usage

```

forest_reg(
  df = NULL,
  uni = NULL,
  multi = NULL,
  desc = NULL,
  theme = NULL,
  ci_col_width = 20,
  side = c("right", "left"),
  quiet = TRUE,
  effects = NULL,
  ticks_at = NULL,
  ticks_digits = NULL,
  xlim = NULL,
  style_strata = TRUE,
  strata_fill = "#EAF2F1",
  ...
)

```

Arguments

<code>df</code>	Output of <code>forest_df()</code> . If <code>NULL</code> , it is built from <code>uni</code> , <code>multi</code> , and <code>desc</code> .
<code>uni</code> , <code>multi</code> , <code>desc</code>	Optional <code>gtregression</code> objects to pass through to <code>forest_df()</code> .
<code>theme</code>	Optional <code>forestploter::forest_theme()</code> . If <code>NULL</code> , <code>forestploter</code> defaults are used. You may pass colors and styling either here or through <code>...</code>
<code>ci_col_width</code>	Numeric value, or one value per effect column, controlling the blank spacer width used by <code>forestploter</code> for the confidence-interval plot column(s). Values greater than 1 are interpreted as approximate character counts. Values between 0 and 1 are accepted for backward compatibility and converted to character counts.

side	Character. For each effect, position of the plot relative to the effect-size text: "left" = plot first then text; "right" = text first then plot. The Characteristic column and descriptive columns remain on the left.
quiet	Logical. Suppress forestploter warnings. Default = 'TRUE'.
effects	Optional effect labels passed to forestploter::forest().
ticks_at	Optional numeric vector, or list of numeric vectors for multiple effect columns, specifying x-axis tick positions. If NULL, forestploter::forest() chooses the default ticks.
ticks_digits	Optional number of digits for x-axis tick labels.
xlim	Optional numeric vector of length 2, or list of length-2 numeric vectors for multiple effect columns, specifying x-axis limits. If NULL, forestploter::forest() chooses the default limits.
style_strata	Logical. Retained for older vertically grouped stratified forest data. Current forest_df() stratified output places strata side by side, so no stratum header rows are styled.
strata_fill	Character. Fill color used for older styled stratum header rows.
...	Passed to forestploter::forest(). Common options include title and footnote.

Value

A gtmregression_forest object with elements:

plot The forest plot object.

data The plotting data sent to forestploter::forest().

input_data The original forest_df() data, including standard-error helper columns.

meta Model metadata, including reference line and x-axis transformation.

Examples

```

birthwt_data <- data_birthwt |>
  transform(
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    ht = factor(ht, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1),
      labels = c("Normal BW", "Low BW"))
  )

uni_or <- uni_reg(
  birthwt_data,
  outcome = "low",
  exposures = c("age", "lwt", "smoke", "ht"),
  approach = "logit"
)

multi_or <- multi_reg(
  birthwt_data,
  outcome = "low",

```

```

    exposures = c("smoke", "ht"),
    adjust_for = c("age", "lwt"),
    approach = "logit"
  )

  forest_reg(uni = uni_or, multi = multi_or)

  # If axis labels overlap, set x-axis limits and tick marks.
  forest_reg(
    uni = uni_or,
    multi = multi_or,
    xlim = list(c(0.25, 8), c(0.25, 12)),
    ticks_at = list(c(0.5, 1, 2, 4, 8), c(0.5, 1, 2, 4, 8))
  )

  # If the forest plot panel is too narrow or too wide, tune ci_col_width.
  forest_reg(
    uni = uni_or,
    multi = multi_or,
    ci_col_width = c(18, 22)
  )

```

gtregression_app

Launch the gtregression app

Description

Open a menu-driven Shiny app for common **gtregression** workflows: data import, descriptive tables, regression tables, survival analysis, diagnostics, causal mediation, plots, and exports.

Usage

```
gtregression_app(..., launch.browser = NULL)
```

Arguments

`...` Additional arguments passed to `shiny::runApp()`.

`launch.browser` Logical; passed to `shiny::runApp()`. The default uses the RStudio Viewer when available, otherwise opens a browser only in interactive sessions.

Details

The app is intentionally kept out of the core package startup path. Shiny and other interface packages are suggested dependencies and are loaded only when `gtregression_app()` is called.

The Advanced tab includes a guided candidate-model builder. Users can name two to six models, choose model-specific exposures and adjustment variables, optionally add an interaction, and track a primary exposure. The app fits compatible `multi_reg()`, `cox_reg()`, or `surv_reg()` objects before passing them to `compare_models()`. Generated code records every fitting call and the final comparison for reproducible use outside the app.

Value

Invisibly returns the result of `shiny::runApp()`.

See Also

`compare_models()`, `multi_reg()`, `cox_reg()`, `surv_reg()`

Examples

```
if (interactive()) {
  gtregression_app()
}
```

identify_confounder	<i>Identify confounders and effect modifiers</i>
---------------------	--

Description

Review whether one or more candidate variables may act as confounders or effect modifiers for one or more exposures.

Usage

```
identify_confounder(
  data,
  outcome = NULL,
  exposure,
  potential_confounder,
  approach = "logit",
  time = NULL,
  event = NULL,
  distribution = "weibull",
  method = "change",
  threshold = 10,
  emm_threshold = 10,
  emm_test = c("interaction", "both", "estimate"),
  interaction_alpha = 0.05,
  format = c("flextable", "gt"),
  theme = c("minimal")
)
```

Arguments

<code>data</code>	A data frame.
<code>outcome</code>	Outcome variable name. Quoted and bare names are accepted. Required for ordinary regression approaches. Leave unset for Cox and parametric survival approaches and supply time and event instead.

exposure	Exposure variable name(s). Can be a character scalar or vector. Quoted names are recommended in scripts, and bare names are also accepted.
potential_confounder	Candidate confounder/effect-modifier variable name(s). Can be a character scalar or vector. Quoted names are recommended in scripts, and bare names are also accepted.
approach	Regression approach. One of "logit", "logbinomial", "poisson", "robpoisson", "linear", "negbin", "cox", or "survreg".
time	Survival time variable name for approach = "cox" or approach = "survreg". Quoted and bare names are accepted.
event	Event indicator variable name for survival approaches. Values may be coded as 0/1, 1/2, logical, or a two-level factor/character variable.
distribution	Parametric survival distribution for approach = "survreg". One of "weibull", "exponential", "lognormal", or "loglogistic".
method	Confounding assessment method. One of "change", "mh", or "both". "change" compares crude and adjusted model estimates. "mh" compares crude and Mantel-Haenszel pooled estimates and is available for binary outcome, binary exposure, and categorical strata. "both" uses either method.
threshold	Percent change threshold for confounding assessment.
emm_threshold	Threshold for relative spread in stratum-specific estimates when using estimate-based effect-modification screening.
emm_test	One of "interaction", "both", or "estimate".
interaction_alpha	Alpha threshold for interaction p-values.
format	Output table format. One of "flextable" (default) or "gt".
theme	Table theme preset or primitives.

Details

The function first assesses possible effect modification using stratum-specific estimates and/or an interaction test. If no important effect modification is detected, it then assesses confounding using the selected method.

This is a screening aid for viewing and organising results. Confounding and effect modification should be interpreted using subject-matter knowledge, study design, and causal diagrams such as DAGs. Automated change-in-estimate and interaction checks should not be used as the sole basis for model adjustment.

For each exposure-candidate pair, the interaction screen fits $\text{outcome} \sim \text{exposure} + \text{potential_confounder} + \text{exposure}:\text{potential_confounder}$. It does not adjust for other covariates. Consequently, the reported interaction p-value is a candidate-only screening result, not a final adjusted interaction analysis. Use `interaction_models(covariates = ...)` when testing a planned, covariate-adjusted interaction.

Use this function when you want to screen one or more candidate variables and organise crude, adjusted, Mantel-Haenszel, and effect-modification signals in one place. For a focused comparison of models with and without a planned exposure-by-modifier interaction term, use `interaction_models()`.

Value

If a single exposure-candidate pair is supplied, returns a detailed list.

If multiple combinations are supplied, returns a list with:

summary A tibble with one row per exposure-candidate combination.

details A named list of detailed results for each combination.

See Also

[interaction_models\(\)](#) for focused model comparison of a planned interaction term.

Examples

```
birthwt_data <- data_birthwt |>
  dplyr::mutate(
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW")),
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    race = factor(race, levels = c(1, 2, 3),
                  labels = c("White", "Black", "Other"))
  )

identify_confounder(
  data = birthwt_data,
  outcome = low,
  exposure = smoke,
  potential_confounder = race,
  approach = logit
)

lung_data <- data_lungcancer |>
  dplyr::mutate(
    trt = factor(trt, levels = c(1, 2),
                 labels = c("Standard treatment", "Test treatment")),
    prior = factor(prior, levels = c(0, 10), labels = c("No", "Yes"))
  )

identify_confounder(
  data = lung_data,
  time = time,
  event = status,
  exposure = trt,
  potential_confounder = prior,
  approach = cox
)
```

interaction_models	<i>Compare Models With and Without an Interaction Term</i>
--------------------	--

Description

Fits two models, one with and one without an interaction term between an exposure and a potential effect modifier. The models are compared using a likelihood ratio test or Wald test to assess statistical evidence of interaction.

Usage

```
interaction_models(
  data,
  outcome = NULL,
  exposure,
  covariates = NULL,
  effect_modifier,
  approach = "logit",
  time = NULL,
  event = NULL,
  distribution = "weibull",
  test = c("LRT", "Wald"),
  alpha = 0.05,
  verbose = FALSE,
  format = c("flextable", "gt", "tibble")
)
```

Arguments

data	A data frame containing all required variables.
outcome	Outcome variable name. Quoted and bare names are accepted. Required for ordinary regression approaches. Leave unset for Cox and parametric survival approaches and supply time and event instead.
exposure	Main exposure variable name. Quoted and bare names are accepted.
covariates	Optional character vector of adjustment covariates. They are included in both the model without and the model with the interaction term. Quoted names are recommended in scripts, and bare names are also accepted.
effect_modifier	Variable name for the potential effect modifier. Quoted and bare names are accepted.
approach	Regression approach. One of "logit", "logbinomial", "poisson", "robpoisson", "negbin", "linear", "cox", or "survreg".
time	Survival time variable name for approach = "cox" or approach = "survreg". Quoted and bare names are accepted.
event	Event indicator variable name for survival approaches.

distribution	Parametric survival distribution for approach = "survreg". One of "weibull", "exponential", "lognormal", or "loglogistic".
test	Statistical test for model comparison. One of "LRT" or "Wald".
alpha	Significance threshold used to classify the interaction result.
verbose	Logical; if TRUE, prints a short interpretation.
format	Output format for the viewing table. One of "flextable" (default), "gt", or "tibble". Use format = "tibble" to keep only the original list structure.

Details

Use this function when the interaction is planned or clinically/causally motivated and you want a focused model comparison. Mantel-Haenszel estimation is not used here because this function tests an explicit interaction term in a regression model. For broader screening of candidate confounders or effect modifiers, including Mantel-Haenszel-supported checks when appropriate, use [identify_confounder\(\)](#).

With covariates = c(age, sex), the two fitted models are outcome ~ exposure + effect_modifier + age + sex and outcome ~ exposure + effect_modifier + age + sex + exposure:effect_modifier. Both models use the same complete-case analysis data, so covariate adjustment is applied consistently to the interaction comparison.

Value

A list with model objects, formulas, p-value, decision, and a one-row summary tibble. When format is "gt" or "flextable", the list also includes table.

See Also

[identify_confounder\(\)](#) for broader confounding and effect-modification screening.

Examples

```
birthwt_data <- data_birthwt |>
  dplyr::mutate(
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW")),
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    race = factor(race, levels = c(1, 2, 3),
                  labels = c("White", "Black", "Other"))
  )

interaction_models(
  data = birthwt_data,
  outcome = low,
  exposure = smoke,
  effect_modifier = race,
  covariates = c(age, lwt),
  approach = logit
)

lung_data <- data_lungcancer |>
```

```

dplyr::mutate(
  trt = factor(trt, levels = c(1, 2),
               labels = c("Standard treatment", "Test treatment")),
  prior = factor(prior, levels = c(0, 10), labels = c("No", "Yes"))
)

interaction_models(
  data = lung_data,
  time = time,
  event = status,
  exposure = trt,
  effect_modifier = prior,
  covariates = c(age, karno),
  approach = cox
)

```

km_plot

Kaplan-Meier survival plot

Description

Plot observed survival over time, with optional confidence intervals, censoring marks, a log-rank p-value, and a number-at-risk table.

Usage

```

km_plot(
  data,
  time,
  event,
  by = NULL,
  conf.int = TRUE,
  risk_table = TRUE,
  p_value = TRUE,
  p_value_position = NULL,
  censor = TRUE,
  break_time_by = NULL,
  xlim = NULL,
  ylim = NULL,
  xlab = "Time",
  ylab = "Survival probability",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  title_size = NULL,
  title_face = "bold",
  legend_title = NULL,

```

```

    legend_position = NULL,
    palette = NULL,
    y_percent = TRUE,
    theme = "classic",
    grid = FALSE,
    base_size = 13
  )

```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time and event status.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>by</code>	Optional grouping variable for separate Kaplan-Meier curves. Quoted and bare names are accepted.
<code>conf.int</code>	Logical; if TRUE, show confidence limits.
<code>risk_table</code>	Logical; if TRUE, add a number-at-risk table below the curve.
<code>p_value</code>	Logical; if TRUE, show the log-rank p-value when <code>by</code> is supplied.
<code>p_value_position</code>	Optional numeric vector of length 2 giving the x and y coordinates for the log-rank p-value inside the plotting panel. If NULL, a lower-left position is chosen automatically.
<code>censor</code>	Logical; if TRUE, show censoring marks.
<code>break_time_by</code>	Optional numeric interval for x-axis and risk-table time breaks. If NULL, breaks are chosen automatically.
<code>xlim</code>	Optional numeric vector of length 2 specifying x-axis limits.
<code>ylim</code>	Optional numeric vector of length 2 specifying y-axis limits. Values may be supplied on the survival-probability scale (for example <code>c(0.5, 1)</code>) or, when <code>y_percent = TRUE</code> , on the percentage scale (for example <code>c(50, 100)</code>).
<code>xlab, ylab</code>	Axis labels.
<code>title</code>	Optional plot title.
<code>subtitle</code>	Optional plot subtitle.
<code>caption</code>	Optional plot caption.
<code>title_size</code>	Optional numeric title font size. If NULL, ggplot2's theme default is used.
<code>title_face</code>	Font face for the title. One of "plain", "bold", "italic", or "bold.italic". Quoted and bare values are accepted.
<code>legend_title</code>	Optional legend title. If NULL, the labelled by variable name is used.
<code>legend_position</code>	Legend position. One of "bottom", "top", "right", "left", or "none". If NULL, grouped plots use "bottom" and ungrouped plots hide the legend. Quoted and bare values are accepted.

palette	Optional character vector of colors for grouped curves.
y_percent	Logical; if TRUE, display survival probability as percentages. If FALSE, display the raw 0 to 1 probability scale.
theme	Plot theme. One of "classic", "minimal", "bw", "light", or "none". Quoted and bare values are accepted.
grid	Logical; if TRUE, show major grid lines. The default is FALSE for a cleaner publication-style Kaplan-Meier plot.
base_size	Base font size.

Value

A ggplot2 object when risk_table = FALSE; otherwise a patchwork object combining the survival curve and risk table. The returned object has attributes fit, plot_data, risk_table, and logrank_p.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                        labels = c("Standard", "Test"))

km_plot(
  data = lung_data,
  time = time,
  event = status
)

km_plot(
  data = lung_data,
  time = time,
  event = status,
  by = trt,
  break_time_by = 200,
  ylim = c(50, 100),
  title = "A. Treatment group",
  title_size = 11,
  legend_position = "none"
)
```

km_risk_table	<i>Kaplan-Meier risk table</i>
---------------	--------------------------------

Description

Tabulate the number at risk at selected follow-up times.

Usage

```
km_risk_table(
  data,
  time,
  event,
  by = NULL,
  times,
  digits = 1,
  extend = TRUE,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time and event status.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>by</code>	Optional grouping variable for separate Kaplan-Meier risk tables. Quoted and bare names are accepted.
<code>times</code>	Numeric vector of follow-up times for the risk table.
<code>digits</code>	Number of digits for displayed follow-up times.
<code>extend</code>	Logical. If TRUE, requested times beyond the observed follow-up range are retained using the last available Kaplan-Meier risk set.
<code>format</code>	Output format. One of "flextable" (default), "gt", or "tibble".
<code>theme</code>	Table styling preset.

Details

The At risk column gives the number still under observation at each requested time. The Events and Censored columns are interval counts up to each requested time point as returned by `summary.survfit()`.

Value

A list of class `c("gtregression", "km_risk_table", ...)` with elements:

`table` A flextable, `gt_tbl`, or NULL when `format = "tibble"`.
`table_body` Tibble with Kaplan-Meier risk table counts.
`table_display` Display data frame used to render the table.
`fit` Fitted `survfit` object.
`time, event, by, times, format, source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                        labels = c("Standard", "Test"))

km_risk_table(
  data = lung_data,
  time = time,
  event = status,
  by = trt,
  times = c(0, 90, 180, 365)
)

km_risk_table(
  data = lung_data,
  time = "time",
  event = "status",
  times = c(0, 90, 180),
  format = tibble
)
```

logrank_test

*Log-rank test for Kaplan-Meier survival curves***Description**

Compare survival curves between groups using the log-rank test.

Usage

```
logrank_test(
  data,
  time,
  event,
  by,
  digits = 2,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

data	A data.frame containing survival time, event status, and grouping variable.
time	Survival follow-up time. Quoted and bare names are accepted.
event	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.

by	Grouping variable used to compare survival curves. Quoted and bare names are accepted.
digits	Number of digits for the chi-square statistic and expected events.
format	Output format. One of "flextable" (default), "gt", or "tibble".
theme	Table styling preset.

Details

The log-rank test compares the observed number of events with the expected number of events in each group under the null hypothesis that the survival curves are the same. It is a group comparison, not an effect-size estimate; use `cox_reg()` when a hazard ratio is needed.

Value

A list of class `c("gtregression", "logrank_test", ...)` with elements:

`table` A flextable, `gt_tbl`, or `NULL` when `format = "tibble"`.

`table_body` Tibble with group-level log-rank components and overall test statistics.

`test` One-row tibble with chi-square statistic, degrees of freedom, and p-value.

`fit` The `survdiff` object.

`time, event, by, format, source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                      labels = c("Standard", "Test"))
```

```
logrank_test(
  data = lung_data,
  time = time,
  event = status,
  by = trt
)
```

```
logrank_test(
  data = lung_data,
  time = "time",
  event = "status",
  by = "trt",
  format = tibble
)
```

mediation_analysis *Causal mediation analysis*

Description

Estimate direct, indirect, total, and proportion mediated effects from regression models.

Usage

```
mediation_analysis(
  data,
  exposure,
  mediator,
  outcome,
  covariates = NULL,
  mediator_approach = "linear",
  outcome_approach = "linear",
  exposure_value = NULL,
  reference_value = NULL,
  sims = 1000,
  conf_level = 0.95,
  seed = NULL,
  format = c("flextable", "gt"),
  theme = c("minimal")
)
```

Arguments

<code>data</code>	A data frame.
<code>exposure</code>	Exposure/treatment variable. Quoted and bare names are accepted.
<code>mediator</code>	Mediator variable. Quoted and bare names are accepted.
<code>outcome</code>	Outcome variable. Quoted and bare names are accepted.
<code>covariates</code>	Optional character vector of covariate names. Quoted names are recommended in scripts, and bare names are also accepted.
<code>mediator_approach</code>	Mediator model. Currently "linear".
<code>outcome_approach</code>	Outcome model. One of "linear" or "logit". Logistic mediation effects are reported on the predicted probability-difference scale.
<code>exposure_value</code>	Exposure value used as the treatment level. If NULL, the second factor level, value 1 for 0/1 variables, or the 75th percentile for continuous exposures is used.
<code>reference_value</code>	Exposure value used as the reference level. If NULL, the first factor level, value 0 for 0/1 variables, or the 25th percentile for continuous exposures is used.
<code>sims</code>	Number of non-parametric bootstrap replicates used for confidence intervals.

conf_level	Confidence level for intervals.
seed	Optional random seed for reproducible bootstrap intervals.
format	One of "flextable" (default) or "gt".
theme	Table theme.

Details

This function is for planned mediation questions, not automatic causal discovery. Causal interpretation requires the usual mediation assumptions, including no unmeasured exposure-outcome, exposure-mediator, or mediator-outcome confounding, correct temporal order, and suitable model specification. Use a directed acyclic graph (DAG) and subject-matter knowledge before interpreting the estimates causally.

For outcome_approach = "linear", effects are mean differences. For outcome_approach = "logit", effects are predicted probability differences, not odds ratios.

Value

A list of class c("gtregression", "mediation_analysis", ...) with elements:

table A formatted flextable or gt_tbl.

table_body Data frame of mediation effect estimates.

table_display Formatted data frame used to build the table.

models Fitted mediator and outcome models.

boot Bootstrap replicate estimates.

values Reference and exposure values used.

variable_labels Named character vector of display labels.

call Matched function call.

Examples

```
med <- mediation_analysis(
  data = data_diabetes_mediation,
  exposure = obesity,
  mediator = glucose,
  outcome = diabetes,
  covariates = c(age, blood_pressure, pregnancies, diabetes_pedigree),
  outcome_approach = logit,
  sims = 50,
  seed = 123
)
```

```
med$table
med$table_body
```

```
# HTML-first output
mediation_analysis(
  data = data_diabetes_mediation,
  exposure = obesity,
```

```

mediator = glucose,
outcome = diabetes,
covariates = c(age, blood_pressure, pregnancies, diabetes_pedigree),
outcome_approach = logit,
format = gt,
sims = 50,
seed = 123
)$table

```

merge_tables

Merge gtregression tables and preserve structure and notes

Description

Merge gtregression tables and preserve structure and notes

Usage

```

merge_tables(
  ...,
  spanners = NULL,
  theme = "minimal",
  format = c("flextable", "gt")
)

```

Arguments

...	Two or more gtregression objects containing \$table_display.
spanners	Character vector of spanner labels, one per table. If NULL, defaults to "Table 1", "Table 2", etc.
theme	Merge theme preset or vector of primitives.
format	Output table format. One of "flextable" (default) or "gt". The merged display is rebuilt in this format, independently of the formats used by the input tables.

Details

Binary variables should use the same row display across all input tables. For a descriptive, uni-variable, and multivariable merge, the clearest publication layout is usually `show_dichotomous = "all_levels"` in `descriptive_table()` and `show_ref = TRUE` in each regression function. Mixing these settings can create additional binary rows; in that situation `merge_tables()` issues a warning with the compatible settings.

Footnotes created by each input table are retained unchanged. Exact duplicate notes are shown once in the merged table.

Value

A merged table object of class `c("gtregression", "merged_table", ...)`.

Examples

```
birthwt_data <- data_birthwt |>
  dplyr::mutate(
    race = factor(race, levels = c(1, 2, 3),
                  labels = c("White", "Black", "Other")),
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    ht = factor(ht, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW"))
  )

uni_tbl <- uni_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("age", "lwt", "smoke", "ht"),
  approach = "logit"
)

multi_tbl <- multi_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("smoke", "ht"),
  adjust_for = c("age", "lwt"),
  approach = "logit"
)

merge_tables(
  uni_tbl,
  multi_tbl,
  spanners = c("Univariable", "Adjusted")
)
```

 modify_table

 Modify Regression/Descriptive Tables (labels, headers, caption, notes)

Description

Works with objects created by this package (class "gtregression"): `uni_reg()`, `multi_reg()`, `cox_reg()`, `surv_reg()`, stratified regression outputs, `descriptive_table()`, and `merge_tables()`.
No **gtsummary** dependency or fallback.

Usage

```
modify_table(
  gt_table,
  variable_labels = NULL,
```

```

    level_labels = NULL,
    header_labels = NULL,
    caption = NULL,
    bold_labels = TRUE,
    bold_levels = FALSE,
    italic_labels = FALSE,
    italic_levels = FALSE,
    remove_N = FALSE,
    remove_N_obs = FALSE,
    remove_abbreviations = FALSE,
    remove_adjustment_note = FALSE,
    caveat = NULL
  )

```

Arguments

<code>gt_table</code>	Table object produced by this package (must contain <code>\$table_display</code>).
<code>variable_labels</code>	Named character vector, for example <code>c(old_var = "New label", ...)</code> .
<code>level_labels</code>	Named list for factor levels: <code>list(var1 = c(old = "New", ...), var2 = c(...))</code> .
<code>header_labels</code>	Named character vector to rename visible headers, e.g. <code>c("OR (95% CI)" = "Crude OR", "p-value" = "P")</code> . Common aliases such as <code>estimate</code> , <code>p.value</code> , and <code>N</code> are also accepted.
<code>caption</code>	Optional caption/title.
<code>bold_labels</code>	Logical; bold variable (header) rows in the body. Defaults to <code>TRUE</code> to preserve the package table hierarchy.
<code>bold_levels</code>	Logical; bold factor level rows in the body.
<code>italic_labels</code>	Logical; italicize variable (header) rows in the body.
<code>italic_levels</code>	Logical; italicize factor level rows in the body.
<code>remove_N</code>	Logical; if <code>TRUE</code> , drops displayed <code>N</code> columns from univariable and stratified package tables. For stratified survival outputs, event columns are retained unless the original table was created with <code>show_sample = "none"</code> .
<code>remove_N_obs</code>	Logical; if <code>TRUE</code> , suppresses multivariable complete-case footnote.
<code>remove_abbreviations</code>	Logical; if <code>TRUE</code> , removes the Abbreviations footnote line.
<code>remove_adjustment_note</code>	Logical; if <code>TRUE</code> , removes the automatic Adjusted for ... footnote. Use <code>caveat</code> to add customised wording.
<code>caveat</code>	Optional extra footnote.

Value

The modified table object (same class as input).

Examples

```

birthwt_data <- data_birthwt |>
  dplyr::mutate(
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    ht = factor(ht, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW"))
  )

tbl <- uni_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("age", "smoke", "ht"),
  approach = "logit",
  format = "gt"
)

modify_table(
  tbl,
  variable_labels = c(age = "Maternal age", smoke = "Smoking"),
  level_labels = list(smoke = c(Yes = "Smoker")),
  header_labels = c(estimate = "Crude OR", p.value = "P"),
  caption = "Univariable regression for low birth weight"
)$table

```

multi_reg

Multivariable regression

Description

Fit adjusted or multivariable regression models and return a manuscript-style table in **flextable** or **gt** format.

Usage

```

multi_reg(
  data,
  outcome,
  exposures,
  adjust_for = NULL,
  interaction = NULL,
  approach = "logit",
  format = c("flextable", "gt"),
  theme = c("minimal"),
  model_stats = FALSE,
  show_ref = TRUE
)

```

Arguments

<code>data</code>	A <code>data.frame</code> containing the variables of interest.
<code>outcome</code>	Character scalar; name of the outcome variable. Quoted and bare names are accepted.
<code>exposures</code>	Character vector; exposure variable(s) to report. Quoted names are recommended in scripts, and bare names are also accepted. If <code>adjust_for = NULL</code> , all exposures are included in a single multivariable model. If <code>adjust_for</code> is supplied, one adjusted model is fitted per exposure and only exposure-specific adjusted estimate(s) are displayed.
<code>adjust_for</code>	Optional character vector of adjustment variables. Quoted and bare names are accepted. Must not overlap with exposures.
<code>interaction</code>	Optional character scalar specifying one interaction term using standard formula syntax, e.g. <code>"bmi*sex"</code> . When used with <code>adjust_for</code> , only a single exposure should be supplied.
<code>approach</code>	Character scalar specifying the regression approach. One of <code>"logit"</code> , <code>"firth"</code> , <code>"logbinomial"</code> , <code>"poisson"</code> , <code>"linear"</code> , <code>"robpoisson"</code> , or <code>"negbin"</code> . Use <code>"firth"</code> for Firth penalized logistic regression, especially with sparse cells or separation.
<code>format</code>	Output table format; one of <code>"flextable"</code> (default) or <code>"gt"</code> .
<code>theme</code>	Table styling preset (e.g. <code>"minimal"</code> , <code>"striped"</code> , <code>"clinical"</code> , <code>"shaded"</code> , <code>"jama"</code>) or a character vector of primitives such as <code>c("plain", "zebra", "lines", "labels_bold", "compact", ...)</code> .
<code>model_stats</code>	Logical; if <code>TRUE</code> , extract model-fit statistics such as AIC, BIC, log-likelihood, deviance, pseudo R-squared for non-linear models, and R-squared for linear models. Statistics are stored in the returned object's <code>model_stats</code> element and are not added to the publication table.
<code>show_ref</code>	Logical; if <code>TRUE</code> (default), display reference-category rows as <code>"Ref."</code> . If <code>FALSE</code> , hide reference rows; a message reminds users to use <code>show_ref = TRUE</code> when reference rows are needed.

Details

Use the default mode when you want all exposures in one model. Use `adjust_for` when you want one adjusted estimate per reported exposure, with the same adjustment set used repeatedly.

If exposure variables have a `"label"` attribute, for example from `labelled::var_label()`, those labels are used automatically in the displayed table and plots. Internal matching still uses the original column names.

In default mode (`adjust_for = NULL`), all exposures are included in a single multivariable model. In adjusted mode, one model is fitted per exposure, adjusting for the variables specified in `adjust_for`.

Interaction terms specified via `interaction` are included in the model using standard formula expansion (e.g. `bmi*sex`). Interaction effects are displayed as additional rows beneath the corresponding exposure.

Value

A list of class `c("gtregression", "multi_reg", ...)` with elements:

table A flextable (if `format="flextable"`) or `gt_tbl` (if `format="gt"`).

table_body A data frame of adjusted estimates and confidence intervals for each exposure and level.

table_display A data frame used for rendering the final table, including header and level rows.

models A list of fitted model(s).

model_summaries `summary()` output for the fitted model(s).

model_stats Model-fit statistics when `model_stats = TRUE`; otherwise `NULL`.

variable_labels Named character vector of display labels used for exposure variables.

reg_check Regression diagnostics for linear models. Printing this element renders a publication-ready table; individual diagnostic rows remain available for programmatic use. Other approaches return a message.

approach The regression approach used.

format The output format used.

source Function identifier (`"multi_reg"`).

adjusted_mode Whether one adjusted model per exposure was fitted.

adjust_for Adjustment variables supplied by the user, if any.

exposures Exposure variables supplied by the user.

interaction Interaction term supplied by the user, if any.

Examples

```
endometrial_data <- data_endometrial
endometrial_data$HG <- factor(endometrial_data$HG, levels = c(0, 1))
endometrial_data$NV <- factor(endometrial_data$NV, levels = c(0, 1))
multi_reg(endometrial_data, HG, c(NV, PI, EH), approach = firth, format = gt)$table
```

plot_mediation

Plot mediation paths

Description

Draw the exposure, mediator, and outcome path from a `mediation_analysis()` result.

Usage

```
plot_mediation(mediation_object, show_estimates = TRUE, base_size = 13)
```

Arguments

mediation_object Object returned by mediation_analysis().

show_estimates Logical; if TRUE, show direct and indirect effect estimates on the plot.

base_size Base font size.

Value

A ggplot2 object.

Examples

```
med <- mediation_analysis(
  data = data_diabetes_mediation,
  exposure = obesity,
  mediator = glucose,
  outcome = diabetes,
  covariates = c(age, blood_pressure, pregnancies, diabetes_pedigree),
  outcome_approach = logit,
  sims = 50,
  seed = 123
)
plot_mediation(med)
plot_mediation(med, show_estimates = FALSE)
```

plot_model_fit	<i>Plot model-fit diagnostics</i>
----------------	-----------------------------------

Description

Visualise model fit for fitted regression models and models stored inside uni_reg() or multi_reg() results.

Usage

```
plot_model_fit(
  model,
  model_name = NULL,
  type = c("auto", "all", "residual", "qq", "scale_location", "cooks",
    "observed_predicted", "calibration"),
  bins = 10,
  base_size = 13
)
```

Arguments

model	A fitted lm or glm model, or a uni_reg() / multi_reg() result.
model_name	Optional model name to select when model contains multiple fitted models. Quoted and bare names are accepted.
type	Plot type. One of "auto", "all", "residual", "qq", "scale_location", "cooks", "observed_predicted", or "calibration". Quoted and bare values are accepted.
bins	Number of groups used for binomial calibration plots.
base_size	Base font size for the plot theme.

Details

plot_model_fit() is a visual check of how a fitted model behaves, not a formal model-selection rule. For survival models, use check_ph() for Cox proportional hazards diagnostics and plot_surv_fit() for parametric survival model fit.

For binomial models, type = "calibration" compares grouped predicted probabilities with observed event proportions. This is most informative for multivariable models, where predictions vary across many patients. A univariable binary predictor may produce only two calibration points; that is expected and simply reflects the two fitted probabilities in the model. Logistic residual plots often show two bands because the outcome is coded as event/non-event.

Value

A ggplot2 object for a single plot, or a patchwork object when multiple diagnostics are requested.

Examples

```
fit_lm <- lm(mpg ~ wt + hp, data = mtcars)
plot_model_fit(fit_lm)

fit_glm <- glm(am ~ mpg + wt, data = mtcars, family = binomial())
plot_model_fit(fit_glm, type = calibration, bins = 4)

uni_fit <- uni_reg(mtcars, am, c(mpg, wt), approach = logit)
plot_model_fit(uni_fit, model_name = mpg, type = residual)
```

plot_reg	<i>Plot regression estimates</i>
----------	----------------------------------

Description

Draw a forest-style plot from a fitted gtre regression object produced by functions such as uni_reg(), multi_reg(), cox_reg(), or surv_reg(). Stratified regression outputs are displayed as one faceted quick-view plot; use forest_df() and forest_reg() for publication-style stratified forest tables.

Usage

```
plot_reg(
  tbl,
  title = NULL,
  caption = NULL,
  ref_line = NULL,
  order_y = NULL,
  log_x = FALSE,
  xlim = NULL,
  breaks = NULL,
  point_color = "#6B7280",
  errorbar_color = "#6B7280",
  point_size = 2.8,
  point_stroke = 0.55,
  ci_linewidth = 0.55,
  base_size = 12,
  show_ref = NULL,
  sig_color = "#0072B2",
  sig_errorbar_color = "#0072B2",
  alpha = 0.05,
  show_adjustment_note = TRUE
)
```

Arguments

<code>tbl</code>	A fitted <code>gtregression</code> object.
<code>title</code>	Optional plot title.
<code>caption</code>	Optional plot caption. If <code>NULL</code> , notes are added automatically for adjusted <code>multi_reg()</code> objects and displayed reference categories.
<code>ref_line</code>	Optional numeric value for the reference line. Defaults to 0 for linear models and 1 otherwise.
<code>order_y</code>	Optional character vector specifying exposure order.
<code>log_x</code>	Logical. If <code>TRUE</code> , use a log-scaled x-axis for ratio measures. The default <code>FALSE</code> uses a linear x-axis.
<code>xlim</code>	Optional numeric vector of length 2 specifying x-axis limits.
<code>breaks</code>	Optional numeric vector of x-axis tick breaks.
<code>point_color</code>	Fill color for non-significant estimate points.
<code>errorbar_color</code>	Color for non-significant confidence intervals.
<code>point_size</code>	Diameter of estimate points in millimetres.
<code>point_stroke</code>	Outline width of estimate points.
<code>ci_linewidth</code>	Line width of confidence intervals.
<code>base_size</code>	Base font size. The default is designed for a clear report or slide figure; increase it for a single large plot.

show_ref	Logical or NULL. The default NULL inherits the reference-row choice used to create tbl. If TRUE, reference rows are shown. If FALSE, binary exposures are shown as compact rows for the estimated non-reference category; affirmative levels such as Yes or 1 are displayed using the variable name.
sig_color	Optional fill color for significant points.
sig_errorbar_color	Optional color for significant error bars.
alpha	Significance level for linear models when p.value is available.
show_adjustment_note	Logical; if TRUE, add a default caption describing adjust_for variables when available.

Value

A ggplot2 object.

plot_reg_combine	<i>Side-by-side forest plots: univariate vs multivariable</i>
------------------	---

Description

Creates two aligned forest plots from compatible gtregrression regression objects such as crude and adjusted outputs from uni_reg(), multi_reg(), cox_reg(), or surv_reg().

Usage

```
plot_reg_combine(
  tbl_uni,
  tbl_multi,
  title_uni = NULL,
  title_multi = NULL,
  caption = NULL,
  ref_line = NULL,
  order_y = NULL,
  log_x = FALSE,
  point_color = "#6B7280",
  errorbar_color = "#6B7280",
  point_size = 2.8,
  point_stroke = 0.55,
  ci_linewidth = 0.55,
  base_size = 12,
  show_ref = NULL,
  sig_color = "#0072B2",
  sig_errorbar_color = "#0072B2",
  xlim_uni = NULL,
  breaks_uni = NULL,
```

```

    xlim_multi = NULL,
    breaks_multi = NULL,
    alpha = 0.05,
    show_adjustment_note = TRUE
  )

```

Arguments

<code>tbl_uni</code>	A univariate gtregrression object.
<code>tbl_multi</code>	A multivariable gtregrression object.
<code>title_uni, title_multi</code>	Optional panel titles.
<code>caption</code>	Optional combined plot caption. If NULL, notes are added automatically for adjusted <code>multi_reg()</code> objects and displayed reference categories.
<code>ref_line</code>	Optional numeric reference line. If NULL, uses 0 for linear models and 1 otherwise.
<code>order_y</code>	Optional character vector to customize exposure ordering.
<code>log_x</code>	Logical. If TRUE, use a log-scaled x-axis for ratio measures. The default FALSE uses a linear x-axis.
<code>point_color, errorbar_color</code>	Base colors for non-significant rows.
<code>point_size</code>	Diameter of estimate points in millimetres.
<code>point_stroke</code>	Outline width of estimate points.
<code>ci_linewidth</code>	Line width of confidence intervals.
<code>base_size</code>	Base font size for the plot theme.
<code>show_ref</code>	Logical or NULL. The default NULL inherits the setting used to create the supplied tables. If TRUE, include reference levels as (Ref.). If FALSE, binary exposures are shown as compact rows for the estimated non-reference category; affirmative levels such as Yes or 1 are displayed using the variable name.
<code>sig_color, sig_errorbar_color</code>	Optional colors for significant rows. If NULL, base colors are reused.
<code>xlim_uni, breaks_uni</code>	Optional x-axis limits and breaks for the univariate panel.
<code>xlim_multi, breaks_multi</code>	Optional x-axis limits and breaks for the multivariable panel.
<code>alpha</code>	Significance level for linear models when <code>p.value</code> is available.
<code>show_adjustment_note</code>	Logical; if TRUE, add a default caption describing <code>adjust_for</code> variables from <code>tbl_multi</code> when available.

Value

A patchwork object with two ggplot2 panels.

plot_surv_fit	<i>Plot observed and fitted parametric survival curves</i>
---------------	--

Description

Compare the observed Kaplan-Meier survival curve with fitted parametric survival curves from `survival::survreg()`.

Usage

```
plot_surv_fit(
  data,
  time,
  event,
  by = NULL,
  adjust_for = NULL,
  distributions = c("weibull", "exponential", "lognormal", "loglogistic"),
  break_time_by = NULL,
  xlim = NULL,
  xlab = "Time",
  ylab = "Survival probability",
  title = NULL,
  legend_title = NULL,
  palette = NULL,
  base_size = 13,
  n_points = 200
)
```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time, event status, and optional grouping or adjustment variables.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>by</code>	Optional grouping variable for observed and fitted curves. Quoted and bare names are accepted.
<code>adjust_for</code>	Optional character vector of adjustment variables included in the fitted parametric model. Fitted curves are predicted at typical adjustment values.
<code>distributions</code>	Parametric survival distributions to overlay. One or more of "weibull", "exponential", "lognormal", or "loglogistic". Quoted and bare values are accepted. Common spellings such as "log-normal" and "log-logistic" are also accepted.
<code>break_time_by</code>	Optional numeric interval for x-axis breaks. If NULL, breaks are chosen automatically.

<code>xlim</code>	Optional numeric vector of length 2 specifying x-axis limits.
<code>xlab, ylab</code>	Axis labels.
<code>title</code>	Optional plot title.
<code>legend_title</code>	Optional legend title. If NULL, the labelled by variable name is used.
<code>palette</code>	Optional character vector of colors for observed groups.
<code>base_size</code>	Base font size.
<code>n_points</code>	Number of points used to draw each fitted curve.

Details

`plot_surv_fit()` is a visual diagnostic for parametric survival modelling. It is useful after `surv_model_compare()` and before treating a final `surv_reg()` model as the preferred model. It is not a Cox-model diagnostic; use `check_ph()` for Cox proportional hazards assumptions.

When `adjust_for` is supplied, fitted curves are predicted at typical adjustment values: medians for numeric variables and the most common level for categorical variables. Use this as a model-fit screen, not as a replacement for clinical or subject-matter judgement.

Value

A `ggplot2` object with attributes `km_fit`, `model_fits`, `observed_data`, `fitted_data`, and `prediction_data`.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                        labels = c("Standard", "Test"))

plot_surv_fit(
  data = lung_data,
  time = time,
  event = status,
  by = trt,
  distributions = c(weibull, lognormal),
  break_time_by = 200
)

plot_surv_fit(
  data = lung_data,
  time = "time",
  event = "status",
  by = "trt",
  adjust_for = c(age, karno),
  distributions = "log-logistic"
)
```

print.gtregression	<i>Print gtregression objects (unified)</i>
--------------------	---

Description

Prints the rendered table for any object produced by this package (objects that include class "gtregression"), regardless of subtype (uni_reg, multi_reg, stratified_*, merged_table, descriptive_table, ...). If no rendered table is found, a compact structure of the object (or its display data) is shown.

Usage

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

Arguments

x	An object with class "gtregression".
...	Ignored. Present for compatibility with the generic.

rmst_table	<i>Restricted mean survival time table</i>
------------	--

Description

Estimate restricted mean survival time (RMST) up to a user-specified follow-up time. RMST is the average survival time observed within a fixed time window, such as 365 days.

Usage

```
rmst_table(
  data,
  time,
  event,
  by = NULL,
  tau,
  digits = 1,
  conf.level = 0.95,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time and event status.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>by</code>	Optional grouping variable for separate RMST estimates. Quoted and bare names are accepted.
<code>tau</code>	Restriction time for RMST. For example, <code>tau = 365</code> reports mean survival time restricted to the first 365 days of follow-up.
<code>digits</code>	Number of digits for displayed survival time summaries.
<code>conf.level</code>	Confidence level for RMST confidence intervals.
<code>format</code>	Output format. One of "flextable" (default), "gt", or "tibble".
<code>theme</code>	Table styling preset.

Details

RMST is estimated from `survival::survfit()` using Kaplan-Meier methods and `summary.survfit(rmean = tau)`. When `by` has exactly two groups, the table includes the RMST difference as the second group minus the first group, with a Wald confidence interval and p-value based on the reported RMST standard errors.

Value

A list of class `c("gtregression", "rmst_table", ...)` with elements:

`table` A flextable, `gt_tbl`, or `NULL` when `format = "tibble"`.

`table_body` Tibble with numeric RMST summaries.

`table_display` Display data frame used to render the table.

`fit` Fitted `survfit` object.

`time, event, by, tau, format, source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                        labels = c("Standard", "Test"))

rmst_table(
  data = lung_data,
  time = time,
  event = status,
  by = trt,
  tau = 365
)
```

```
rmst_table(
  data = lung_data,
  time = "time",
  event = "status",
  tau = 180,
  format = tibble
)
```

save_docx

Save multiple tables and plots to a Word document

Description

Saves a collection of gtregression tables, merged tables, gt_tbl/flextable objects, and ggplot2 plots into a single Word document.

Usage

```
save_docx(
  tables = NULL,
  plots = NULL,
  filename = "report.docx",
  titles = NULL,
  table_width = 6.5,
  plot_width = 6,
  plot_height = 5
)
```

Arguments

tables	A list of tables. Each element may be a gtregression object, merged_table object, gt_tbl, or flextable.
plots	A list of ggplot2 plot objects.
filename	File name for the output, with or without .docx. If no directory is supplied, the file is saved in tempdir().
titles	Optional character vector of titles for tables and plots in the order they are added.
table_width	Maximum table width in inches for Word export. The default 6.5 fits a standard portrait Word page with common margins. Use NULL to keep the original flextable widths.
plot_width	Width of inserted plots in inches.
plot_height	Height of inserted plots in inches.

Value

Saves the Word document to disk. Invisibly returns the normalized file path.

Examples

```
birthwt_data <- data_birthwt |>
  dplyr::mutate(
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW"))
  )

tbl <- uni_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("age", "smoke"),
  approach = "logit",
  format = "flectable"
)

save_docx(tables = tbl, filename = tempfile("report"))
```

save_forest

Save a forest_reg() output

Description

Saves a forest_reg() output, or a compatible **forestploter**/grid object, to a fixed graphics device. This is useful when the RStudio Viewer or operating-system graphics device crops wide forest plots or compresses forest columns.

Usage

```
save_forest(
  forest,
  filename = "forest",
  format = c("pdf", "png", "tiff", "jpg"),
  width = NULL,
  height = NULL,
  scale = 1,
  auto_size = TRUE,
  padding = 0.25,
  dpi = 300
)
```

Arguments

forest	A gtregrression_forest object returned by forest_reg(), or a compatible forestploter /grid object.
filename	File name for the output, with or without extension. If no directory is supplied, the file is saved in tempdir().
format	Output format. One of "pdf", "png", "tiff", or "jpg".

width, height	Optional export width and height in inches. If either is NULL and auto_size = TRUE, the size is measured from the intrinsic forestploter gtable dimensions. If measurement is not possible, a practical fallback is estimated from the number of rows and columns in the forest_reg() output.
scale	Positive multiplier applied to the export width and height. This is a quick way to make a large forest plot roomier.
auto_size	Logical. If TRUE, measure the intrinsic forest plot size when width or height is not supplied. If FALSE, use the fallback row/column-based sizing.
padding	White space around the forest plot in inches.
dpi	Resolution for raster formats.

Value

Saves the file to disk. Invisibly returns the normalized file path.

Examples

```

birthwt_data <- data_birthwt |>
  transform(
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    ht = factor(ht, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW"))
  )

uni_or <- uni_reg(
  birthwt_data,
  outcome = "low",
  exposures = c("age", "smoke", "ht"),
  approach = "logit"
)

forest <- forest_reg(uni = uni_or)
save_forest(forest, filename = tempfile("forest"), format = "pdf")

# For large forest plots, increase width, height, scale, or padding.
save_forest(
  forest,
  filename = tempfile("forest-wide"),
  format = "png",
  scale = 1.2,
  padding = 0.35,
  dpi = 300
)

```

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. If no directory is supplied, the file is saved in tempdir().
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.

Value

Saves the file to disk. Invisibly returns the normalized file path.

Examples

```
p <- ggplot2::ggplot(mtcars, ggplot2::aes(wt, mpg)) +  
  ggplot2::geom_point()  
  
save_plot(p, filename = tempfile("plot"), format = "png")
```

save_table	<i>Save a single regression or summary table</i>
------------	--

Description

Saves a gtregression table, merged table, gt_tbl, or flextable as a Word, RTF, PDF, or HTML file.

Usage

```
save_table(
  tbl,
  filename = "table",
  format = c("docx", "rtf", "pdf", "html"),
  orientation = c("auto", "portrait", "landscape"),
  fit_width = TRUE,
  font_size = 9,
  min_font_size = 8
)
```

Arguments

<code>tbl</code>	A gtregrression object, merged_table object, gt_tbl, or flextable.
<code>filename</code>	File name for the output. Extension is optional. If no directory is supplied, the file is saved in <code>tempdir()</code> .
<code>format</code>	Output format. One of "docx", "rtf", "pdf", or "html". Flextable objects can be saved as "docx", "rtf", or "html".
<code>orientation</code>	Word page orientation for DOCX output. One of "auto", "portrait", or "landscape". With "auto", wide tables are saved in landscape orientation before any font-size reduction is attempted.
<code>fit_width</code>	Logical. If TRUE, try to fit flextable DOCX output within the selected Word page width. If FALSE, keep the natural autofit table width.
<code>font_size</code>	Requested font size for flextable DOCX output.
<code>min_font_size</code>	Smallest font size allowed when fitting wide flextable DOCX output. The font size is never reduced below this value.

Value

Saves the file to disk. Invisibly returns the normalized file path.

Examples

```
birthwt_data <- data_birthwt |>
  dplyr::mutate(
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1), labels = c("Normal BW", "Low BW"))
  )

tbl <- uni_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("age", "smoke"),
  approach = "logit"
)

save_table(tbl, filename = tempfile("table"), format = "html")
```

```
# Wide Word tables can be saved in landscape orientation.

save_table(
  tbl,
  filename = tempfile("table-wide"),
  format = "docx",
  orientation = "auto",
  fit_width = TRUE,
  font_size = 9,
  min_font_size = 8
)
```

select_models

Stepwise model selection with fit metrics

Description

Compare candidate models added or removed stepwise, then return the accepted steps, model-fit metrics, and the lowest-AIC model.

Usage

```
select_models(
  data,
  outcome,
  exposures,
  approach = "logit",
  time = NULL,
  event = NULL,
  distribution = "weibull",
  direction = "forward",
  format = c("flextable", "gt", "tibble")
)
```

Arguments

data	A data frame containing the outcome and predictor variables.
outcome	A single character string indicating the outcome variable. Quoted and bare names are accepted. Not used for survival approaches when time and event are supplied.
exposures	Character vector of predictor variables to consider. Quoted names are recommended in scripts, and bare names are also accepted.
approach	Regression method. One of: "logit", "logbinomial", "poisson", "robpoisson", "negbin", "linear", "cox", or "survreg".
time, event	Survival time and event indicator for approach = "cox" or approach = "survreg". Quoted and bare names are accepted.

distribution	Parametric survival distribution for approach = "survreg". One of "weibull", "exponential", "lognormal", or "loglogistic". Common spellings such as "log-normal" and "log-logistic" are also accepted.
direction	Stepwise selection direction. One of: "forward" (default), "backward", or "both".
format	Output format for the viewing table. One of "flextable" (default), "gt", or "tibble". Use format = "tibble" to keep only the original list structure.

Details

Treat stepwise selection as a screening tool. It is best used alongside the study question, clinical judgement, and model diagnostics rather than as an automatic final-model rule.

Value

A list with the following components:

- `results_table`: A tibble summarising each accepted step's model metrics (AIC, BIC, deviance, log-likelihood, and adjusted R-squared for linear models).
- `best_model`: The best-fitting model object based on lowest AIC.
- `all_models`: A named list of the accepted stepwise models.
- `direction`: Stepwise selection direction used.
- `table`: A formatted `gt_tbl` or flextable when format is "gt" or "flextable".

Examples

```
data <- data_birthwt
stepwise <- select_models(
  data = data,
  outcome = "bwt",
  exposures = c("age", "lwt", "smoke"),
  approach = "linear",
  direction = "forward"
)
stepwise$results_table
stepwise$best_model
```

stratified_multi_reg *Stratified multivariable regression*

Description

Fits multivariable regression models within each stratum and returns a unified wide table with one "Characteristic" column and, under bold spanners for each stratum, model N, "Adjusted <effect>", and "p-value".

Usage

```
stratified_multi_reg(
  data,
  outcome,
  exposures,
  stratifier,
  adjust_for = NULL,
  interaction = NULL,
  approach = "logit",
  format = c("flextable", "gt"),
  theme = c("minimal"),
  show_ref = TRUE
)
```

Arguments

<code>data</code>	A data frame containing the variables.
<code>outcome</code>	Character scalar; name of the outcome variable. Quoted and bare names are accepted.
<code>exposures</code>	Character vector of exposure variables to report. Quoted names are recommended in scripts, and bare names are also accepted.
<code>stratifier</code>	Character scalar; name of the stratifying variable. Quoted and bare names are accepted.
<code>adjust_for</code>	Optional character vector of adjustment variables. Quoted and bare names are accepted. This argument works the same way as in <code>multi_reg()</code> .
<code>interaction</code>	Optional character scalar specifying one interaction term using standard formula syntax, e.g. <code>"bmi*sex"</code>
<code>approach</code>	One of <code>"logit"</code> , <code>"firth"</code> , <code>"logbinomial"</code> , <code>"poisson"</code> , <code>"linear"</code> , <code>"robpoisson"</code> , or <code>"negbin"</code>
<code>format</code>	One of <code>"flextable"</code> (default) or <code>"gt"</code> .
<code>theme</code>	Preset name (e.g. <code>"minimal"</code> , <code>"striped"</code> , <code>"clinical"</code> , <code>"shaded"</code> , <code>"jama"</code>) or primitives <code>c("plain", "zebra", "lines", "labels_bold", "compact", "header_shaded")</code>
<code>show_ref</code>	Logical; if <code>TRUE</code> (default), display reference-category rows as <code>"Ref."</code> . If <code>FALSE</code> , hide reference rows; a message reminds users to use <code>show_ref = TRUE</code> when reference rows are needed.

Details

If `adjust_for = NULL`, all exposures are included in one multivariable model within each stratum. If `adjust_for` is supplied, one adjusted model is fitted per exposure within each stratum.

If exposure variables have a `"label"` attribute, for example from `labelled::var_label()`, those labels are used automatically in the displayed table. Internal matching still uses the original column names.

Value

A list of class `c("gtregression", "stratified_multi_reg", ...)` with:

`table` A flextable (format = "flextable") or gt_tbl (format = "gt").

`table_display` Wide data frame used to build the table.

`per_stratum` Named list of per-stratum regression results.

`models` Named list of fitted models by stratum.

`model_summaries` Named list of model summaries by stratum.

`variable_labels` Named character vector of display labels used for exposure variables.

`reg_check` Named list of diagnostics by stratum.

`by, levels, approach, format, source` Metadata fields.

Examples

```
birthwt_data <- data_birthwt |>
  transform(
    race = factor(race, levels = c(1, 2, 3),
                  labels = c("White", "Black", "Other")),
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    ht = factor(ht, levels = c(0, 1), labels = c("No", "Yes")),
    ui = factor(ui, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1),
                  labels = c("Normal BW", "Low BW"))
  )

stratified_multi <- stratified_multi_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("age", "lwt", "smoke", "ht"),
  stratifier = "race",
  approach = "logit",
  format = "gt"
)

stratified_adjusted <- stratified_multi_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("smoke", "ht", "ui"),
  stratifier = "race",
  adjust_for = c("age", "lwt"),
  approach = "logit",
  format = "gt"
)
```

stratified_uni_reg	<i>Stratified univariable regression</i>
--------------------	--

Description

Fit univariable regression models within each level of a stratifier and combine the results into a stratified table rendered with **gt** or **flextable**.

Usage

```
stratified_uni_reg(
  data,
  outcome,
  exposures,
  stratifier,
  approach = "logit",
  format = c("flextable", "gt"),
  theme = c("minimal"),
  show_ref = TRUE
)
```

Arguments

<code>data</code>	A data frame containing the variables.
<code>outcome</code>	Character scalar; name of the outcome variable. Quoted and bare names are accepted.
<code>exposures</code>	Character vector of exposure variables to model. Quoted names are recommended in scripts, and bare names are also accepted.
<code>stratifier</code>	Character scalar; name of the stratifying variable. Quoted and bare names are accepted.
<code>approach</code>	Modeling approach. One of "logit", "firth", "logbinomial", "poisson", "robpoisson", "linear", or "negbin".
<code>format</code>	Output table format; one of "flextable" (default) or "gt".
<code>theme</code>	Table styling preset or theme primitives.
<code>show_ref</code>	Logical; if TRUE (default), display reference-category rows as "Ref.". If FALSE, hide reference rows; a message reminds users to use <code>show_ref = TRUE</code> when reference rows are needed.

Details

If exposure variables have a "label" attribute, for example from `labelled::var_label()`, those labels are used automatically in the displayed table. Internal matching still uses the original column names.

Value

A list of class `c("gtregression", "stratified_uni_reg", ...)` with elements:

- `table` A rendered `gt_tbl` or `flextable`.
- `table_display` Display-ready wide stratified results.
- `per_stratum` List of complete `uni_reg()` results by stratum.
- `models` List of fitted model objects by stratum.
- `model_summaries` List of model summaries by stratum.
- `variable_labels` Named character vector of display labels used for exposure variables.
- `reg_check` Regression diagnostics by stratum.
- `by` The stratifier variable.
- `levels` Strata included in the analysis.
- `approach` The regression approach used.
- `format` The output format used.
- `source` Function identifier (`"stratified_uni_reg"`).

Accessors

- `$table` Rendered stratified regression table.
- `$table_display` Wide display data used to build the table.
- `$per_stratum` Full `uni_reg()` result objects by stratum.
- `$models` List of fitted model objects for each stratum.
- `$model_summaries` List of model summaries.
- `$reg_check` Diagnostic check results (when applicable).

See Also

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

Examples

```
birthwt_data <- data_birthwt |>
  transform(
    race = factor(race, levels = c(1, 2, 3),
                  labels = c("White", "Black", "Other")),
    smoke = factor(smoke, levels = c(0, 1), labels = c("No", "Yes")),
    low = factor(low, levels = c(0, 1),
                 labels = c("Normal BW", "Low BW"))
  )

stratified_uni <- stratified_uni_reg(
  data = birthwt_data,
  outcome = "low",
  exposures = c("age", "lwt", "smoke"),
  stratifier = "race",
```

```
    approach = "logit",
    format = "gt"
  )

  stratified_uni$table
```

survival_prob	<i>Kaplan-Meier survival probability table</i>
---------------	--

Description

Estimate Kaplan-Meier survival probabilities at user-specified follow-up times, such as 30-day, 6-month, or 1-year survival.

Usage

```
survival_prob(
  data,
  time,
  event,
  by = NULL,
  times,
  digits = 1,
  extend = TRUE,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

data	A data.frame containing survival time and event status.
time	Survival follow-up time. Quoted and bare names are accepted.
event	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
by	Optional grouping variable for separate Kaplan-Meier survival probabilities. Quoted and bare names are accepted.
times	Numeric vector of follow-up times at which survival probability should be estimated.
digits	Number of digits for percentages and survival probabilities.
extend	Logical. If TRUE, requested times beyond the observed follow-up range are retained using the last available Kaplan-Meier estimate.
format	Output format. One of "flextable" (default), "gt", or "tibble".
theme	Table styling preset.

Details

Survival probabilities are estimated from `survival::survfit()` at the requested follow-up times. Events and censored counts are interval counts up to each requested time point as returned by `summary.survfit()`.

Value

A list of class `c("gtregression", "survival_prob", ...)` with elements:

`table` A flextable, `gt_tbl`, or `NULL` when `format = "tibble"`.

`table_body` Tibble with Kaplan-Meier survival probabilities.

`table_display` Display data frame used to render the table.

`fit` Fitted `survfit` object.

`time, event, by, times, format, source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                        labels = c("Standard", "Test"))
```

```
survival_prob(
  data = lung_data,
  time = time,
  event = status,
  by = trt,
  times = c(90, 180, 365)
)
```

```
survival_prob(
  data = lung_data,
  time = "time",
  event = "status",
  times = c(90, 180),
  format = tibble
)
```

survival_quantiles	<i>Kaplan-Meier survival quantile table</i>
--------------------	---

Description

Estimate Kaplan-Meier survival time quantiles, such as the 25th percentile, median, and 75th percentile survival times.

Usage

```
survival_quantiles(
  data,
  time,
  event,
  by = NULL,
  probs = c(0.25, 0.5, 0.75),
  digits = 1,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time and event status.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>by</code>	Optional grouping variable for separate Kaplan-Meier quantiles. Quoted and bare names are accepted.
<code>probs</code>	Numeric vector of event-time quantiles to estimate. The default <code>c(0.25, 0.5, 0.75)</code> reports the 25th percentile, median, and 75th percentile event times.
<code>digits</code>	Number of digits for survival time summaries.
<code>format</code>	Output format. One of "flextable" (default), "gt", or "tibble".
<code>theme</code>	Table styling preset.

Details

A probability of 0.50 is the median event time: the estimated time by which 50 A probability of 0.25 is the time by which 25 corresponding to 75 follow-up, the display table shows "Not reached".

Value

A list of class `c("gtregression", "survival_quantiles", ...)` with elements:

`table` A flextable, `gt_tbl`, or `NULL` when `format = "tibble"`.
`table_body` Tibble with Kaplan-Meier quantiles.
`table_display` Display data frame used to render the table.
`fit` Fitted `survfit` object.
`time, event, by, probs, format, source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                        labels = c("Standard", "Test"))

survival_quantiles(
  data = lung_data,
  time = time,
  event = status,
  by = trt
)

survival_quantiles(
  data = lung_data,
  time = "time",
  event = "status",
  probs = c(0.25, 0.5),
  format = tibble
)
```

survival_summary	<i>Kaplan-Meier survival summary table</i>
------------------	--

Description

Summarise Kaplan-Meier follow-up with total N, events, censoring, and median survival with a 95

Usage

```
survival_summary(
  data,
  time,
  event,
  by = NULL,
  digits = 1,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

data	A data.frame containing survival time and event status.
time	Survival follow-up time. Quoted and bare names are accepted.
event	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.

<code>by</code>	Optional grouping variable for separate Kaplan-Meier summaries. Quoted and bare names are accepted.
<code>digits</code>	Number of digits for survival time summaries.
<code>format</code>	Output format. One of "flextable" (default), "gt", or "tibble".
<code>theme</code>	Table styling preset.

Details

Median survival is estimated from `survival::survfit()`. If the median survival time is not reached during follow-up, the display table shows "Not reached".

Value

A list of class `c("gtregression", "survival_summary", ...)` with elements:

`table` A flextable, `gt_tbl`, or `NULL` when `format = "tibble"`.

`table_body` Tibble with numeric Kaplan-Meier summaries.

`table_display` Display data frame used to render the table.

`fit` Fitted `survfit` object.

`time, event, by, format, source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
  labels = c("Standard", "Test"))
```

```
survival_summary(
  data = lung_data,
  time = time,
  event = status,
  by = trt
)
```

```
survival_summary(
  data = lung_data,
  time = "time",
  event = "status",
  format = tibble
)
```

surv_model_compare	<i>Compare parametric survival model distributions</i>
--------------------	--

Description

Fit the same parametric survival regression model using multiple distributions and compare model-fit statistics such as AIC and BIC.

Usage

```
surv_model_compare(
  data,
  time,
  event,
  exposures,
  adjust_for = NULL,
  distributions = c("weibull", "exponential", "lognormal", "loglogistic"),
  digits = 2,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time, event status, and predictor variables.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>exposures</code>	Character vector of main exposure variable names. Quoted names are recommended in scripts, and bare names are also accepted.
<code>adjust_for</code>	Optional character vector of adjustment variables. These variables are included in every candidate model.
<code>distributions</code>	Parametric survival distributions to compare. Defaults to <code>c("weibull", "exponential", "lognormal", "loglogistic")</code> . Quoted and bare values are accepted. Common spellings such as "log-normal" and "log-logistic" are also accepted.
<code>digits</code>	Number of digits for displayed model statistics.
<code>format</code>	Output format. One of "flextable" (default), "gt", or "tibble".
<code>theme</code>	Table styling preset.

Details

The same model formula is fitted for each candidate distribution using `survival::survreg()`. Lower AIC or BIC values indicate better relative model fit among the compared distributions. These statistics should be used with clinical judgment and visual checks; they do not prove that a distribution is scientifically correct.

Value

A list of class `c("gtregression", "surv_model_compare", ...)` with elements:

`table` A flextable, `gt_tbl`, or `NULL` when `format = "tibble"`.

`table_body` Tibble with model-fit statistics.

`table_display` Display data frame used to render the table.

`models` Named list of fitted survreg models.

`time, event, exposures, adjust_for, distributions, format, source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                      labels = c("Standard", "Test"))
lung_data$prior <- factor(lung_data$prior, levels = c(0, 10),
                        labels = c("No", "Yes"))

surv_model_compare(
  data = lung_data,
  time = time,
  event = status,
  exposures = c("trt", "celltype", "prior"),
  adjust_for = c("age", "karno")
)

surv_model_compare(
  data = lung_data,
  time = "time",
  event = "status",
  exposures = c(trt, prior),
  distributions = c(weibull, "log-logistic"),
  format = tibble
)
```

surv_predict

Predict survival probabilities from a parametric survival model

Description

Estimate model-based survival probabilities at user-specified follow-up times from a fitted `survival::survreg()` model.

Usage

```
surv_predict(
  model,
  newdata = NULL,
  times,
  digits = 1,
  format = c("flextable", "gt", "tibble"),
  theme = c("minimal")
)
```

Arguments

<code>model</code>	A fitted survreg model, or a <code>surv_reg()</code> object containing exactly one fitted model.
<code>newdata</code>	Optional <code>data.frame</code> of profiles for prediction. If <code>NULL</code> , one typical profile is built from the model data using medians for numeric variables and the most common level for categorical variables.
<code>times</code>	Numeric vector of follow-up times at which survival probability should be predicted.
<code>digits</code>	Number of digits for displayed follow-up times and probabilities.
<code>format</code>	Output format. One of "flextable" (default), "gt", or "tibble".
<code>theme</code>	Table styling preset.

Details

`surv_predict()` is for parametric survival regression models fitted by `surv_reg()` or `survival::survreg()`. It is not a Kaplan-Meier estimate and it is not a Cox prediction helper.

Supported distributions are "weibull", "exponential", "lognormal", and "loglogistic", matching `surv_reg()`. Predictions are conditional on the supplied profile and the chosen parametric distribution.

Value

A list of class `c("gtregression", "surv_predict", ...)` with elements:

`table` A flextable, `gt_tbl`, or `NULL` when `format = "tibble"`.

`table_body` Tibble with predicted survival probabilities.

`table_display` Display data frame used to render the table.

`model` The fitted survreg model used for prediction.

`newdata`, `times`, `distribution`, `format`, `source` Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
  labels = c("Standard", "Test"))
```

```
fit <- surv_reg(  
  data = lung_data,  
  time = time,  
  event = status,  
  exposures = "trt",  
  adjust_for = c("age", "karno"),  
  distribution = weibull  
)  
  
surv_predict(  
  model = fit$models$trt,  
  newdata = data.frame(  
    trt = factor("Test", levels = levels(lung_data$trt)),  
    age = 60,  
    karno = 70  
  ),  
  times = c(90, 180, 365)  
)  
  
surv_predict(  
  model = fit,  
  times = c(90, 180),  
  format = tibble  
)
```

surv_reg*Parametric survival regression*

Description

Fit parametric survival models and report time ratios.

Usage

```
surv_reg(  
  data,  
  time,  
  event,  
  exposures,  
  adjust_for = NULL,  
  stratifier = NULL,  
  interaction = NULL,  
  multivariable = FALSE,  
  multivariate = NULL,  
  distribution = "weibull",  
  format = c("flextable", "gt"),  
  theme = c("minimal"),
```

```

    show_sample = "events",
    model_stats = FALSE,
    show_ref = TRUE
  )

```

Arguments

<code>data</code>	A <code>data.frame</code> containing survival time, event status, and exposure variables.
<code>time</code>	Survival follow-up time. Quoted and bare names are accepted.
<code>event</code>	Event indicator. Quoted and bare names are accepted. Numeric 0/1, numeric 1/2, logical, character, and factor variables are accepted. For two-level character or factor variables, the second level is treated as the event.
<code>exposures</code>	Character vector of exposure variable names. Quoted names are recommended in scripts, and bare names are also accepted.
<code>adjust_for</code>	Optional character vector of adjustment variables. When supplied, one adjusted model is fitted per exposure.
<code>stratifier</code>	Optional single stratifying variable. When supplied, stratum-specific parametric survival tables are produced using the same crude, adjusted, or multivariable workflow requested by the other arguments. The stratifier cannot also be used as the time, event, exposure, adjustment, or interaction variable.
<code>interaction</code>	Optional character scalar specifying one interaction term using standard formula syntax, e.g. <code>"trt*prior"</code> . Quoted and bare interaction syntax are accepted. In exposure-by-exposure mode, supply a single exposure; in multivariable = TRUE mode, the interaction is added to the single multivariable model.
<code>multivariable</code>	Logical; if FALSE (default), the current exposure-by-exposure workflow is used. If TRUE, one multivariable parametric survival model is fitted using all variables in exposures, and all exposure coefficients are reported.
<code>multivariate</code>	Optional logical alias for multivariable. This is accepted for convenience; multivariable is used internally.
<code>distribution</code>	Parametric survival distribution. One of <code>"weibull"</code> , <code>"exponential"</code> , <code>"lognormal"</code> , or <code>"loglogistic"</code> . Quoted and bare values are accepted. Common spellings such as <code>"log-normal"</code> and <code>"log-logistic"</code> are also accepted.
<code>format</code>	Output table format; one of <code>"flectable"</code> (default) or <code>"gt"</code> .
<code>theme</code>	Table styling preset.
<code>show_sample</code>	For stratified parametric survival tables, controls which sample-size columns are shown in the publication table. One of <code>"events"</code> (default), <code>"n"</code> , <code>"both"</code> , or <code>"none"</code> . Model statistics, when requested, still retain both N and event counts.
<code>model_stats</code>	Logical; if TRUE, extract model-fit statistics including AIC, BIC, log-likelihood, scale, number of events, and N.
<code>show_ref</code>	Logical; if TRUE (default), display reference-category rows as <code>"Ref."</code> . If FALSE, hide reference rows; a message reminds users to use <code>show_ref = TRUE</code> when reference rows are needed.

Details

`surv_reg()` fits accelerated failure time style parametric survival models using `survival::survreg()`. The exponentiated coefficient is displayed as a time ratio. A time ratio above 1 suggests longer survival time; a time ratio below 1 suggests shorter survival time, conditional on the chosen distribution.

By default, `surv_reg()` keeps the exposure-by-exposure workflow: without `adjust_for`, one crude model is fitted per exposure; with `adjust_for`, one adjusted model is fitted per exposure and only the exposure estimate is reported.

With `multivariable = TRUE`, all variables in exposures are included in one parametric survival model and all coefficients are reported. Since these estimates are adjusted for the other variables in the same model, the table reports Adjusted Time Ratio (95% CI). The `adjust_for` argument is not used in this mode; include every variable that belongs in the model inside exposures.

Interaction terms specified via `interaction` are included using standard formula expansion (for example, `trt*prior`). Interaction effects are displayed as additional rows beneath the corresponding exposure.

Stratified parametric survival tables show event counts by default. Use `show_sample = "n"`, `show_sample = "both"`, or `show_sample = "none"` to control the displayed sample columns. Crude stratified tables calculate these counts for each exposure-specific model; adjusted and multivariable stratified tables use the corresponding fitted model within each stratum.

If exposure variables have a "label" attribute, for example from `labelled::var_label()`, those labels are used automatically in the displayed table.

Value

A list of class `c("gtregression", "surv_reg", ...)` with elements:

table A flextable or `gt_tbl`.

table_body Data frame of time ratios and confidence intervals.

table_display Data frame used to render the publication table.

models List of fitted `survreg` models.

model_summaries Summary output for the fitted models.

model_stats Model-fit statistics when `model_stats = TRUE`; otherwise `NULL`.

variable_labels Named character vector of display labels.

footnotes Character vector used by the rendered table. Adjustment notes use the same display labels as the table.

time,event,distribution,approach,format,source,adjust_for,exposures,interaction Metadata fields.

Examples

```
lung_data <- data_lungcancer
lung_data$trt <- factor(lung_data$trt, levels = c(1, 2),
                      labels = c("Standard", "Test"))
lung_data$prior <- factor(lung_data$prior, levels = c(0, 10),
                        labels = c("No", "Yes"))
```

```

surv_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c("trt", "celltype", "karno", "age")
)

surv_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c(trt, celltype, prior),
  adjust_for = c(age, karno),
  distribution = lognormal
)

# Interaction in an adjusted exposure model
surv_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = trt,
  adjust_for = c(age, karno),
  interaction = trt*prior,
  distribution = weibull
)

surv_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c(trt, celltype, prior, age, karno),
  distribution = weibull,
  multivariable = TRUE
)

# multivariate is accepted as an alias
surv_reg(
  data = lung_data,
  time = time,
  event = status,
  exposures = c(trt, age, karno),
  multivariate = TRUE
)

```

uni_reg

Univariate regression

Description

Fit one model per exposure and return a clean regression table in **flextable** or **gt** format.

Usage

```
uni_reg(
  data,
  outcome,
  exposures,
  approach = "logit",
  format = c("flextable", "gt"),
  theme = c("minimal"),
  model_stats = FALSE,
  show_ref = TRUE
)
```

Arguments

<code>data</code>	A data frame containing the outcome and exposure variables.
<code>outcome</code>	Character scalar; outcome column name. Quoted and bare names are accepted.
<code>exposures</code>	Character vector; exposure column names. Quoted names are recommended in scripts, and bare names are also accepted.
<code>approach</code>	Regression approach. One of "logit", "firth", "logbinomial", "poisson", "robpoisson", "linear", or "negbin". Use "firth" for Firth penalized logistic regression, especially with sparse cells or separation.
<code>format</code>	One of "flextable" (default) or "gt".
<code>theme</code>	Preset name (e.g. "minimal", "striped", "clinical", "shaded", "jama") or primitives <code>c("plain", "zebra", "lines", "labels_bold", "compact", "header_shaded")</code>
<code>model_stats</code>	Logical; if TRUE, extract model-fit statistics such as AIC, BIC, log-likelihood, deviance, pseudo R-squared for non-linear models, and R-squared for linear models. Statistics are stored in the returned object's <code>model_stats</code> element and are not added to the publication table.
<code>show_ref</code>	Logical; if TRUE (default), display reference-category rows as "Ref.". If FALSE, hide reference rows; a message reminds users to use <code>show_ref = TRUE</code> when reference rows are needed.

Details

Use this when you want a quick crude association table before building an adjusted model. The fitted models are kept in the returned object, so the formatted table does not hide the underlying analysis.

If exposure variables have a "label" attribute, for example from `labelled::var_label()`, those labels are used automatically in the displayed table and plots. Internal matching still uses the original column names.

Value

A list of class `c("gtregression", "uni_reg", ...)` with elements:

table A flextable (when `format="flextable"`) or `gt_tbl` (when `format="gt"`).

table_body Data frame of numeric estimates and CIs.

table_display Data frame for display (headers + levels).

models List of fitted univariate models.

model_summaries Per-model summary() results.

model_stats Model-fit statistics when model_stats = TRUE; otherwise NULL.

variable_labels Named character vector of display labels used for exposure variables.

reg_check Diagnostics for linear models. Printing this element renders a publication-ready table; individual diagnostic rows remain available for programmatic use. Other approaches return a message.

approach, format, source Metadata fields.

Examples

```
d <- mtcars
if (requireNamespace("gt", quietly = TRUE)) {
  uni_reg(d, "am", c("mpg", "cyl"), approach = "logit", format = "gt")$table
}
if (requireNamespace("flextable", quietly = TRUE)) {
  ft <- uni_reg(d, "am", c("mpg", "cyl"), approach = "logit", format = "flextable")
  class(ft$table)
}

endometrial_data <- data_endometrial
endometrial_data$HG <- factor(endometrial_data$HG, levels = c(0, 1))
endometrial_data$NV <- factor(endometrial_data$NV, levels = c(0, 1))
uni_reg(endometrial_data, HG, c(NV, PI, EH), approach = "firth", format = "gt")$table
```

\$.gtregression	<i>Access fields on gtregression objects with '\$'</i>
-----------------	--

Description

Works for any object from this package, since they all carry class "gtregression". Returns NULL (quietly) if the field is not present.

Usage

```
## S3 method for class 'gtregression'
x$name
```

Arguments

x	A gtregression object.
name	Field name to access.

Details

Common fields: - table, table_display, table_body - models, model_summaries, reg_check - approach, format (or engine), source - parts, spanners (for merged tables) - by, levels (for descriptive tables)

Index

* datasets

- data_birthwt, [12](#)
- data_diabetes_mediation, [13](#)
- data_endometrial, [14](#)
- data_epilepsy, [15](#)
- data_gt_quin, [16](#)
- data_infertility, [16](#)
- data_lungcancer, [17](#)
- data_SynthDiabetes, [18](#)
- \$.gtregression, [77](#)

- check_collinearity, [3](#)
- check_convergence, [4](#)
- check_ph, [6](#)
- compare_models, [7](#)
- cox_reg, [9](#)

- data_birthwt, [12](#)
- data_diabetes_mediation, [13](#)
- data_endometrial, [14](#)
- data_epilepsy, [15](#)
- data_gt_quin, [16](#)
- data_infertility, [16](#)
- data_lungcancer, [17](#)
- data_SynthDiabetes, [18](#)
- descriptive_table, [18](#)
- dissect, [20](#)

- forest_df, [21](#)
- forest_reg, [22](#)

- gtregression_app, [24](#)

- identify_confounder, [25](#), [29](#)
- interaction_models, [26](#), [27](#), [28](#)

- km_plot, [30](#)
- km_risk_table, [32](#)

- logrank_test, [34](#)

- mediation_analysis, [36](#)
- merge_tables, [38](#)
- modify_table, [39](#)
- multi_reg, [41](#)

- plot_mediation, [43](#)
- plot_model_fit, [44](#)
- plot_reg, [45](#)
- plot_reg_combine, [47](#)
- plot_surv_fit, [49](#)
- print.gtregression, [51](#)

- rmst_table, [51](#)

- save_docx, [53](#)
- save_forest, [54](#)
- save_plot, [55](#)
- save_table, [56](#)
- select_models, [58](#)
- stratified_multi_reg, [59](#)
- stratified_uni_reg, [62](#)
- surv_model_compare, [69](#)
- surv_predict, [70](#)
- surv_reg, [72](#)
- survival_prob, [64](#)
- survival_quantiles, [65](#)
- survival_summary, [67](#)

- uni_reg, [75](#)