Package: rifttable (via r-universe)

October 31, 2026

Type Package

Title Results Tables to Bridge the Rift Between Epidemiologists and Their Data

Version 0.7.3

Description Presentation-ready results tables for epidemiologists in an automated, reproducible fashion. The user provides the final analytical dataset and specifies the design of the table, with rows and/or columns defined by exposure(s), effect modifier(s), and estimands as desired, allowing to show descriptors and inferential estimates in one table -- bridging the rift between epidemiologists and their data, one table at a time. See Rothman (2017) <doi:10.1007/s10654-017-0314-3>.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.1.0)

Imports broom (>= 0.7.0), dplyr (>= 1.0.8), purrr, risks (>= 0.4.3), rlang (>= 0.4.0), stats, survival, stringr, tibble, tidyr

Suggests gt (>= 0.8.0), knitr, markdown, quantreg, rmarkdown, sandwich, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://stopsack.github.io/rifttable/, https://github.com/stopsack/rifttable/

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Konrad H. Stopsack [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-0722-1311>)

Maintainer Konrad H. Stopsack <stopsack@post.harvard.edu>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-31 17:50:11 UTC

RemoteUrl https://github.com/cran/rifttable

RemoteRef HEAD

RemoteSha 86da38bf487c18e3fb9cee260e355e4aad4b7e78

Contents

rifttable 2
rt_gt 10
scoreci 12
survdiff_ci 13
table1_design 15

Index **17**

`rifttable` *Results Tables for Epidemiology*

Description

This function displays descriptive and inferential results for binary, continuous, and survival data in the format of a table stratified by exposure and, if requested, by effect modifiers.

This function is intended only for tabulations of final results. Model diagnostics for regression models need to be conducted separately.

Usage

```
rifttable(
  design,
  data,
  id = "",
  layout = "rows",
  factor = 1000,
  risk_percent = FALSE,
  risk_digits = dplyr::if_else(risk_percent == TRUE, true = 0, false = 2),
  diff_digits = 2,
  ratio_digits = 2,
  ratio_digits_decrease = c(`2.995` = -1, `9.95` = -2),
  rate_digits = 1,
  to = ", ",
  reference = "(reference)",
  type2_layout = "rows",
  overall = FALSE,
  exposure_levels = c("noempty", "nona", "all")
)
```

Arguments

design	Design matrix (data frame) that sets up the table. See Details. Must be provided.
data	Dataset to be used for all analyses. Must be provided unless the design was generated by table1_design.
id	Optional. Name of an id variable in the data that identifies clustered observations, for example if the data are in a long format with rows encoding time-varying covariates. See vignettes for which estimators use this information. Defaults to "", i.e., each row is a unique individual.
layout	Optional. "rows" uses the design as rows and exposure categories as columns. "cols" is the opposite: design as columns and exposure categories as rows. Defaults to "rows".
factor	Optional. Used for type = "rates": Factor to multiply events per person-time by. Defaults to 1000.
risk_percent	Optional. Show risk and risk difference estimates in percentage points instead of proportions. Defaults to FALSE unless the design was generated by table1_design. In this latter case, if risk_percent is not provided, it will default to TRUE.
risk_digits	Optional. Number of decimal digits to show for risks/ cumulative incidence. Defaults to 2 for risk_percent = FALSE and to 0 for risk_percent = TRUE. Alternatively, digits can be specified directly for each row of the design.
diff_digits	Optional. Number of decimal digits to show for rounding of means and mean difference estimates. Defaults to 2. Alternatively, digits can be specified directly for each row of the design.
ratio_digits	Optional. Number of decimal digits to show for ratio estimates. Defaults to 2. Alternatively, digits can be specified directly for each row of the design.
ratio_digits_decrease	Optional. Lower limits of ratios above which fewer digits should be shown. Provide a named vector of the format, c(`3` = -1, `10` = -2) to reduce the number of rounding digits by 1 digit for ratios greater than 3 and by 2 digits for ratios greater than 10 (the default). To disable, set to NULL.
rate_digits	Optional. Number of decimal digits to show for rates. Defaults to 1. Alternatively, digits can be specified directly for each row of the design.
to	Optional. Separator between the lower and the upper bound of the 95% confidence interval (and interquartile range for medians). Defaults to ", ".
reference	Optional. Defaults to "(reference)". Alternative label for the reference category.
type2_layout	Optional. If a second estimate is requested via type2 in the design matrix, display it as rows below ("rows") or as columns ("columns") to the right. Defaults to "rows".
overall	Optional. Defaults to FALSE. Add a first column with unstratified estimates to an exposure-stratified table? Elements will be shown only for absolute estimates (e.g., type = "mean") and blank for comparative estimates (e.g., mean difference via type = "diff").

exposure_levels
: Optional. Defaults to "noempty". Show only exposure levels that exist in the data or are NA ("noempty"); show only exposure levels that are neither NA nor empty ("nona"); or show all exposure levels even if they are NA or a factor level that does not occur in the data ("all").

Details

The main input parameter is the dataset design. Always required are the column type (the type of requested statistic, see below), as well as outcome for binary outcomes or time and event for survival outcomes:

- label A label for each row (or column). If missing, type will be used as the label.
- exposure Optional. The exposure variable. Must be categorical (factor or logical). If missing (NA), then an unstratified table with absolute estimates only will be returned.
- outcome The outcome variable for non-survival data (i.e., whenever event and time are not used). For risk/prevalence data, this variable must be 0/1 or FALSE/TRUE.
- time The time variable for survival data. Needed for, e.g., type = "hr" and type = "rate" (i.e., whenever outcome is not used).
- time2 The second time variable for late entry models. Only used in conjunction with time. If provided, time will become the entry time and time2 the exit time, following conventions of Surv.
- event The event variable for survival data. Events are typically 1, censored observations 0. If competing events are present, censoring should be the first-ordered level, e.g., of a factor, and the level corresponding to the event of interest should be supplied as event = "event_variable@Recurrence" if "Recurrence" is the event of interest. The event variable is needed for, e.g., type = "hr" and type = "rate", i.e., whenever outcome is not used.
- trend Optional. For regression models, a continuous representation of the exposure, for which a slope per one unit increase ("trend") will be estimated. Must be a numeric variable. If joint models for exposure and effect_modifier are requested, trends are still reported within each stratum of the effect_modifier. Use NA to leave blank.
- effect_modifier Optional. A categorical effect modifier variable. Use NA to leave blank.
- stratum Optional. A stratum of the effect modifier. Use NULL to leave blank. NA will evaluate observations with missing data for the effect_modifier. The spelling strata is also accepted as a fallback.
- confounders Optional. A string in the format "+ var1 + var2" that will be substituted into into formula = exposure + confounders. Use NA or "" (empty string) to leave blank; the default. For Cox models, can add "+ strata(site)" to obtain models with stratification by, e.g., site. For Poisson models, can add "+ offset(log(persontime))" to define, e.g., persontime as the offset.
- weights Optional. Variable with weights, for example inverse- probability weights. Used by comparative survival estimators (e.g., type = "hr" and type = "cumincdiff") as well as type = "cuminc" and type = "surv". They are ignored by other estimators. The spelling weight is also accepted as a fallback.
- type The statistic requested (case-insensitive):

  Comparative estimates with 95% confidence intervals:

- "hr" Hazard ratio from Cox proportional hazards regression.
- "irr" Incidence rate ratio for count outcomes from Poisson regression model.
- "irrrob" Ratio for other outcomes from Poisson regression model with robust (sandwich) standard errors.
- "rr" Risk ratio (or prevalence ratio) from riskratio. Can request specific model fitting approach and, for marginal standardization only, the number of bootstrap repeats. Examples: "rrglm_start" or "rrmargstd 2000".
- "rd" Risk difference (or prevalence difference) from riskdiff. Can request model fitting approach and bootstrap repeats as for "rr".
- "diff" Mean difference from linear model.
- "quantreg" Quantile difference from quantile regression using rq with method = "fn". By default, this is the difference in medians. For a different quantile, e.g., the 75th percentile, use "quantreg 0.75".
- "fold" Fold change from generalized linear model with log link (i.e., ratio of arithmetic means).
- "foldlog" Fold change from linear model after log transformation of the outcome (i.e., ratio of geometric means).
- "or" Odds ratio from logistic regression.
- "survdiff" Difference in survival from Kaplan-Meier estimator. Provide time horizon, e.g., "survdiff 2.5" to evaluate differences in survival at 2.5 years. Uses survdiff_ci.
- "cumincdiff" Difference in cumulative incidence from the Kaplan-Meier estimator or, if competing risks are present, its generalized form, the Aalen-Johansen estimator. Provide time horizon, e.g., "cumincdiff 2.5" to evaluate differences in cumulative incidence at 2.5 years. Uses survdiff_ci.
- "survratio" Ratio in survival from Kaplan-Meier estimator. Provide time horizon, e.g., "survdiff 2.5" to evaluate 2.5-year relative risk. Uses survdiff_ci.
- "cumincratio" Ratio in cumulative incidence from the Kaplan-Meier estimator or, if competing risks are present, its generalized form, the Aalen-Johansen estimator. Provide time horizon, e.g., "cumincdiff 2.5" to evaluate the 2.5-year risk difference. Uses survdiff_ci.

Absolute estimates per exposure category:

- "events" Event count.
- "time" Person-time.
- "outcomes" Outcome count.
- "total" Number of observations.
- "events/time" Events slash person-time.
- "events/total" Events slash number of observations.
- "cases/controls" Cases and non-cases (events and non-events); useful for case-control studies.
- "risk" Risk (or prevalence), calculated as a proportion, i.e., outcomes divided by number of observations. Change between display as proportion or percent using the parameter risk_percent.
- "risk (ci)" Risk with 95% confidence interval (Wilson score interval for binomial proportions, see scoreci).

- "cuminc" Cumulative incidence ("risk") from the Kaplan-Meier estimator or, if competing risks are present, its generalized form, the Aalen-Johansen estimator. Provide time point (e.g., 1.5-year cumulative incidence) using "cuminc 1.5". If no time point is provided, the cumulative incidence at end of follow-up is returned. Change between display as proportion or percent using the parameter risk_percent.
- "cuminc (ci)" Cumulative incidence ("risk"), as above, with 95% confidence intervals (Greenwood standard errors with log transformation, the default of the survival package/survfit). Provide time point as in "cuminc".
- "surv" Survival from the Kaplan-Meier estimator. Provide time point (e.g., 1.5-year survival) using "surv 1.5". If no time point is provided, returns survival at end of follow-up. Change between display as proportion or percent using the parameter risk_percent.
- "surv (ci)" Survival from the Kaplan-Meier estimator with 95% confidence interval (Greenwood standard errors with log transformation, the default of the survival package/survfit). Provide time point as in "surv".
- "rate" Event rate: event count divided by person-time, multiplied by factor.
- "rate (ci)" Event rate with 95% confidence interval (Poisson-type large-sample interval).
- "outcomes (risk)" A combination: Outcomes followed by risk in parentheses.
- "outcomes/total (risk)" A combination: Outcomes slash total followed by risk in parentheses.
- "events/time (rate)" A combination: Events slash time followed by rate in parentheses.
- "medsurv" Median survival.
- "medsurv (ci)" Median survival with 95% confidence interval.
- "medfu" Median follow-up (reverse Kaplan-Meier), equals median survival for censoring.
- "medfu (iqr)" Median and interquartile range for follow-up.
- "maxfu" Maximum follow-up time.
- "mean" Mean (arithmetic mean).
- "mean (ci)" Mean and 95% CI.
- "mean (sd)" Mean and standard deviation.
- "geomean" Geometric mean.
- "median" Median.
- "median (iqr)" Median and interquartile range.
- "range" Range: Minimum to maximum value.
- "sum" Sum.
- "blank" or "" An empty line.
- Custom: A custom function that must be available under the name estimate_my_function in order to be callable as type = "my_function".

By default, regression models will be fit separately for each stratum of the effect_modifier. Append "_joint" to "hr", "rr", "rd", "irr", "irrrob", "diff", "fold", "foldlog", "quantreg", or "or" to obtain "joint" models for exposure and effect modifier that have a single reference category. Example: type = "hr_joint". The reference categories for exposure and effect modifier are their first factor levels, which can be changed using fct_relevel from the forcats package. Note that the joint model will be fit across all non-missing (NA) strata of the effect modifier, even if the design table does not request all strata be shown.

- type2 Optional. A second statistic that is added in an adjacent row or column (global option type2_layout defaults to "row" and can alternatively be set to "column"). For example, use type = "events/times", type2 = "hr" to get both event counts/person-time and hazard ratios for the same data, exposure, stratum, confounders, and outcome.
- timepoint Optional. The time point at which a time-to-event statistic should be estimated. Used by type (estimators): "cuminc", "surv", "cumincdiff", "cumincratio", "survdiff", "survratio"; ignored otherwise.
- digits Optional. The number of digits for rounding an individual line. Defaults to NA, where the number of digits will be determined based on rifttable's arguments risk_percent, risk_digits, diff_digits, ratio_digits, or rate_digits, as applicable.
- digits2 Optional. As digits, for the second estimate (type2).
- nmin Optional. Minimum count per cell below which data should be suppressed. For example, if there are only 3 events in an exposure category and nmin = 5, then "-- (<5)" will be printed for that cell. The nmin cut-off will be applied to the number of events for survival-type outcomes, to the number of observations with the outcome for binary outcomes, or to the number of observations for continuous outcomes (and all other estimators). Defaults to NA, i.e., to return all estimates.
- na_rm Optional. Exclude observations with missing outcome. Defaults to FALSE. Use with caution.
- ci Optional. Confidence level. Defaults to 0.95.
- arguments Optional additional arguments for some estimators (type), e.g., for type = "rr". See vignettes.

Use tibble, tribble, and mutate to construct the design dataset, especially variables that are used repeatedly (e.g., exposure, time,event, or outcome). See examples.

If regression models cannot provide estimates in a stratum, e.g., because there are no events, then "--" will be printed. Accompanying warnings need to be suppressed manually, if appropriate, using suppressWarnings(rifttable(...)).

Value

Tibble. Get formatted output as a gt table by passing on to rt_gt.

References

Greenland S, Rothman KJ (2008). Introduction to Categorical Statistics. In: Rothman KJ, Greenland S, Lash TL. Modern Epidemiology, 3rd edition. Philadelpha, PA: Lippincott Williams & Wilkins. Page 242. (Poisson/large-sample approximation for variance of incidence rates)

Examples

```
# Load 'cancer' dataset from survival package (Used in all examples)
data(cancer, package = "survival")

# The exposure (here, 'sex') must be categorical
cancer <- cancer |>
  tibble::as_tibble() |>
  dplyr::mutate(
```

```
    sex = factor(
      sex,
      levels = 1:2,
      labels = c("Male", "Female")
    ),
    time = time / 365.25,
    status = status - 1
  )


# Example 1: Binary outcomes (use 'outcome' variable)
# Set table design
design1 <- tibble::tibble(
  label = c(
    "Outcomes",
    "Total",
    "Outcomes/Total",
    "Risk",
    "Risk (CI)",
    "Outcomes (Risk)",
    "Outcomes/Total (Risk)",
    "RR",
    "RD"
  )
) |>
  dplyr::mutate(
    type = label,
    exposure = "sex",
    outcome = "status"
  )

# Generate rifttable
rifttable(
  design = design1,
  data = cancer
)

# Use 'design' as columns (selecting RR and RD only)
rifttable(
  design = design1 |>
    dplyr::filter(label %in% c("RR", "RD")),
  data = cancer,
  layout = "cols"
)


# Example 2: Survival outcomes (use 'time' and 'event'),
#   with an effect modifier and a confounder
# Set table design
design2 <- tibble::tribble(
  # Elements that vary by row:
  ~label,                      ~stratum, ~confounders, ~type,
  "**Overall**",               NULL,     "",           "blank",
```

```
  "  Events",                   NULL,     "",          "events",
  "  Person-years",             NULL,     "",          "time",
  "  Rate/1000 py (95% CI)",    NULL,     "",          "rate (ci)",
  "  Unadjusted HR (95% CI)",   NULL,     "",          "hr",
  "  Age-adjusted HR (95% CI)", NULL,     "+ age",     "hr",
  "",                           NULL,     "",          "blank",
  "**Stratified models**",      NULL,     "",          "",
  "*ECOG PS1* (events/N)",      1,        "",          "events/total",
  "  Unadjusted",               1,        "",          "hr",
  "  Age-adjusted",             1,        "+ age",     "hr",
  "*ECOG PS2* (events/N)",      2,        "",          "events/total",
  "  Unadjusted",               2,        "",          "hr",
  "  Age-adjusted",             2,        "+ age",     "hr",
  "",                           NULL,     "",          "",
  "**Joint model**, age-adj.",  NULL,     "",          "",
  "  ECOG PS1",                 1,        "+ age",     "hr_joint",
  "  ECOG PS2",                 2,        "+ age",     "hr_joint"
) |>
  # Elements that are the same for all rows:
  dplyr::mutate(
    exposure = "sex",
    event = "status",
    time = "time",
    effect_modifier = "ph.ecog"
  )

# Generate rifttable
rifttable(
  design = design2,
  data = cancer |>
    dplyr::filter(ph.ecog %in% 1:2)
)


# Example 3: Get two estimates using 'type' and 'type2'
design3 <- tibble::tribble(
  ~label,     ~stratum, ~type,          ~type2,
  "ECOG PS1", 1,        "events/total", "hr",
  "ECOG PS2", 2,        "events/total", "hr"
) |>
  dplyr::mutate(
    exposure = "sex",
    event = "status",
    time = "time",
    confounders = "+ age",
    effect_modifier = "ph.ecog"
  )

rifttable(
  design = design3,
  data = cancer |>
    dplyr::filter(ph.ecog %in% 1:2)
)
```

```
rifttable(
  design = design3,
  data = cancer |>
    dplyr::filter(ph.ecog %in% 1:2),
  layout = "cols",
  type2_layout = "cols"
)


# Example 4: Continuous outcomes (use 'outcome' variable);
# request rounding to 1 decimal digit in some cases;
# add continuous trend (slope per one unit of the 'trend' variable)
tibble::tribble(
  ~label,                   ~stratum, ~type,        ~digits,
  "Marginal mean (95% CI)", NULL,     "mean (ci)",  1,
  "  Male",                 "Male",   "mean",       NA,
  "  Female",               "Female", "mean",       NA,
  "",                       NULL,     "",           NA,
  "Stratified model",       NULL,     "",           NA,
  "  Male",                 "Male",   "diff",       1,
  "  Female",               "Female", "diff",       1,
  "",                       NULL,     "",           NA,
  "Joint model",            NULL,     "",           NA,
  "  Male",                 "Male",   "diff_joint", NA,
  "  Female",               "Female", "diff_joint", NA
) |>
  dplyr::mutate(
    exposure = "ph.ecog_factor",
    trend = "ph.ecog",
    outcome = "age",
    effect_modifier = "sex"
  ) |>
  rifttable(
    data = cancer |>
      dplyr::filter(ph.ecog < 3) |>
      dplyr::mutate(ph.ecog_factor = factor(ph.ecog))
  )


# Example 5: Get formatted output for Example 2
rifttable(
  design = design2,
  data = cancer |>
    dplyr::filter(ph.ecog %in% 1:2)
) |>
  rt_gt()
```

---

rt_gt *Turn tibble into gt Table with Custom Formatting*

Description

Formatting includes:

- Text align to top/left
- Smaller row padding
- No top border
- Bold column labels

If this function is called within a document that is being knit to plain markdown, such as `format: gfm` in a Quarto document or `format: github_document` in an RMarkdown document, then a plain markdown-formatted table (e.g., without footnotes) is returned via `kable`.

Usage

```
rt_gt(df, md = 1, indent = 10, remove_border = TRUE)
```

Arguments

`df`	Data frame/tibble
`md`	Optional. If not `NULL`, then the given columns will be printed with markdown formatting, e.g., `md = c(1, 3)` for columns 1 and 3. Defaults to `1`, i.e., the first column.
`indent`	Optional. Detects cells in the first column of table, e.g., from `rifttable` where the first column contains the labels, that start with at least two spaces. This text is then indented via `tab_style`. Defaults `10` for 10 pixels. Set to `NULL` to turn off.
`remove_border`	Optional. For rows that are indented in the first column or have an empty first column, remove the upper horizontal border line? Defaults to `TRUE`.

Value

Formatted gt table

Examples

```
data(mtcars)
mtcars |>
  dplyr::slice(1:5) |>
  rt_gt()
```

scoreci *Wilson Score Confidence Intervals*

Description

"This function computes a confidence interval for a proportion. It is based on inverting the large-sample normal score test for the proportion." (Alan Agresti, who wrote the original R code)

Inputs for `success`, `total`, and `level` are vectorized.

Usage

```
scoreci(success, total, level = 0.95, return_midpoint = FALSE)
```

Arguments

`success`	Success count.
`total`	Total count.
`level`	Optional. Confidence level. Defaults to 0.95.
`return_midpoint`	Optional. Return midpoint of confidence interval? Defaults to `FALSE`.

Value

Data frame:

- `success` Success count
- `total` Total count
- `estimate` Proportion
- `conf.low` Lower bound of the confidence interval.
- `conf.high` Upper bound of the confidence interval.
- `midpoint` Mid-point of the confidence interval (for `return_midpoint = TRUE`).
- `level` Confidence level.

See Also

https://users.stat.ufl.edu/~aa/cda/R/one-sample/R1/index.html

Agresti A, Coull BA. Approximate is better than "exact" for interval estimation of binomial proportions. Am Stat 1998;52:119-126. doi:10.2307/2685469

Brown LD, Cai TT, DasGupta A. Interval estimation for a binomial proportion (with discussion). Stat Sci 2001;16:101-133. doi:10.1214/ss/1009213286

Examples

```
scoreci(success = 5, total = 10)
scoreci(success = c(5:10), total = 10, level = 0.9)
```

survdiff_ci *Estimate Difference in Survival or Cumulative Incidence and Confidence Interval*

Description

This function estimates the unadjusted difference or ratio in survival or cumulative incidence (risk) at a given time point based on the difference between per-group Kaplan-Meier estimates or, if competing events are present, Aalen-Johansen estimates of the cumulative incidence.

For constructing confidence limits, the MOVER approach described by Zou and Donner (2008) is used, with estimation on the log scale for ratios.

Usage

```
survdiff_ci(
  formula,
  data,
  time,
  estimand = c("survival", "cuminc"),
  type = c("diff", "ratio"),
  approach = c("mover", "squareadd"),
  conf.level = 0.95,
  event_type = NULL,
  id_variable = NULL,
  weighted = FALSE
)
```

Arguments

formula	Formula of a survival object using Surv of the form, Surv(time, event) ~ group. The exposure variable (here, group) must be categorical with at least 2 categories.
data	Data set.
time	Time point to estimate survival difference at.
estimand	Optional. Estimate difference in survival ("survival") or cumulative incidence ("cuminc")? This parameter affects the sign of the differences. Only "cuminc" is available if competing events are present, i.e., event_type is not NULL. Defaults to "survival".
type	Optional. Estimate differences ("diff") or ratio ("ratio") of survival or cumulative incidence? Defaults to "diff".
approach	Optional. For estimating confidence limits of differences, use the MOVER approach based on upper and lower confidence limits of each group ("mover"), or square-and-add standard errors ("squareadd")? Defaults to "mover". (For confidence limits of ratios, this argument is ignored and MOVER is used.)
conf.level	Optional. Confidence level. Defaults to 0.95.

event_type	Optional. Event type (level) for event variable with competing events. Defaults to NULL.
id_variable	Optional. Identifiers for individual oberversations, required if data are clustered, or if competing events and time/time2 notation are used concomitantly.
weighted	Optional. Weigh survival curves, e.g. for inverse-probability weighting, before estimating differences or ratios? If TRUE, the data must contain a variable called .weights. Defaults to FALSE.

Value

Tibble in tidy format:

- term Name of the exposure stratum.
- estimate Difference or ratio.
- std.error Large-sample standard error of the difference in survival functions (see References). For each survival function, Greenwood standard errors with log transformation are used, the default of the survival package/survfit).
- statistic z statistic.
- p.value From the z statistic, two-sided.
- conf.low Lower confidence limit
- conf.high Upper confidence limit

References

Com-Nougue C, Rodary C, Patte C. How to establish equivalence when data are censored: a randomized trial of treatments for B non-Hodgkin lymphoma. Stat Med 1993;12:1353–64. doi:10.1002/sim.4780121407

Altman DG, Andersen PK. Calculating the number needed to treat for trials where the outcome is time to an event. BMJ 1999;319:1492–5. doi:10.1136/bmj.319.7223.1492

Zou GY, Donner A. Construction of confidence limits about effect measures: A general approach. Statist Med 2008;27:1693–1702. doi:10.1002/sim.3095

Examples

```
# Load 'cancer' dataset from survival package (Used in all examples)
data(cancer, package = "survival")

cancer <- cancer |>
  dplyr::mutate(
    sex = factor(
      sex,
      levels = 1:2,
      labels = c("Male", "Female")
    ),
    status = status - 1
  )
```

```
survdiff_ci(
  formula = survival::Surv(time = time, event = status) ~ sex,
  data = cancer,
  time = 365.25
)
# Females have 19 percentage points higher one-year survival than males
# (95% CI, 5 to 34 percentage points).
```

`table1_design` *Design A Descriptive Table*

Description

This function generates a `design` table from which `rifttable` can generate a descriptive table.

Usage

```
table1_design(
  data,
  ...,
  by = NULL,
  total = TRUE,
  empty_levels = FALSE,
  na_always = FALSE,
  na_label = "Unknown",
  continuous_type = "median (iqr)",
  binary_type = "outcomes (risk)"
)
```

Arguments

`data`	Data set
`...`	Optional: Variables to include or exclude (using `-variable`)
`by`	Optional: Stratification variable. Typically the exposure.
`total`	Optional: Whether to add the total count at the beginning. Defaults to `TRUE`.
`empty_levels`	Optional: Whether to include empty levels of factor variables. Defaults to `FALSE`.
`na_always`	Optional: Whether to add the count of missing values for each variable, even if there are none. Defaults to `FALSE`, i.e., the count of missing values will only be shown if there are any.
`na_label`	Label for count of missing values. Defaults to `"Unknown"`.
`continuous_type`	Estimator (`type` in `rifttable` `design`) for continuous variables. Defaults to `"median (iqr)"`.
`binary_type`	Estimator (`type` in `rifttable` `design`) for binary variables and strata of categorical variables. Defaults to `"outcomes (risk)"` (count and column proportion).

Value

design tibble that can be passed on to rifttable. Contains an attribute rt_data so that the dataset does not have to be provided to rifttable another time.

Examples

```
# Data preparation
cars <- tibble::as_tibble(mtcars) |>
  dplyr::mutate(
    gear = factor(
      gear,
      levels = 3:5,
      labels = c("Three", "Four", "Five")
    ),
    # Categorical version of "hp", shows each category
    hp_categorical = dplyr::if_else(
      hp >= 200,
      true = "200+ hp",
      false = "<200 hp"
    ),
    # Binary version of "hp", shows the TRUEs
    hp_binary = hp >= 200
  )
# Label some variables. Better alternative: labelled::set_variable_labels()
attr(cars$hp, "label") <- "Horsepower"
attr(cars$hp_categorical, "label") <- "Horsepower"
attr(cars$hp_binary, "label") <- "200+ hp"
attr(cars$am, "label") <- "Automatic transmission"
attr(cars$gear, "label") <- "Gears"

# Generate table "design"
design <- cars |>
  table1_design(
    hp, hp_categorical, hp_binary, mpg, am,
    by = gear
  )

# Use "design" to create a descriptive table.
design |>
  rifttable(diff_digits = 0)

# Obtain a formatted table
design |>
  rifttable(diff_digits = 0) |>
  rt_gt()
```

Index

kable, *11*

mutate, *7*

rifttable, 2, *11*, *15, 16*
riskdiff, *5*
riskratio, *5*
rq, *5*
rt_gt, *7*, 10

scoreci, *5*, 12
Surv, *4*, *13*
survdiff_ci, *5*, 13
survfit, *6*, *14*

tab_style, *11*
table1_design, *3*, 15
tibble, *7*
tidy, *14*
tribble, *7*