Package 'gtsummary'

August 20, 2019

Title Presentation-Ready Data Summary and Analytic Result Tables

Version 1.2.1

Description Creates presentation-ready tables summarizing data sets, regression models, and more. The code to create the tables is concise and highly customizable. Data frames can be summarized with any function, e.g. mean(), median(), even user-written functions. Regression models are summarized and include the reference rows for categorical variables. Common regression models, such as logistic regression and Cox proportional hazards regression, are automatically identified and the tables are pre-filled with appropriate column headers. The package is enhanced when the 'gt' package is installed. Use this code to install: 'remotes::install_github(``rstudio/gt'')'.

License MIT + file LICENSE

URL http://www.danieldsjoberg.com/gtsummary/

BugReports https://github.com/ddsjoberg/gtsummary/issues

Depends R (>= 3.5)

Imports broom (>= 0.5.1), broom.mixed (>= 0.2.3), crayon (>= 1.3.4), dplyr (>= 0.7.8), glue (>= 1.3.0), knitr (>= 1.21), magrittr (>= 1.5), purrr (>= 0.3.0), rlang (>= 0.3.1), stringr (>= 1.3.1), survival, tibble (>= 2.0.1), tidyr (>= 0.8.2), tidyselect (>= 0.2.5)

Suggests car (>= 3.0.2), covr (>= 3.2.1), curl (>= 3.3), geepack (>= 1.2.1), ggplot2 (>= 3.1.0), Hmisc (>= 4.2.0), lme4 (>= 1.1.18.1), remotes (>= 2.1.0), rmarkdown (>= 1.11), spelling (>= 2.0), testthat (>= 2.1.0), usethis (>= 1.5.0)

Enhances gt (>= 0.1.0)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Daniel D. Sjoberg [aut, cre] (<https://orcid.org/0000-0003-0862-2018>),
Margie Hannum [aut] (<https://orcid.org/0000-0002-2953-0449>),
Karissa Whiting [aut] (<https://orcid.org/0000-0002-4683-1868>),
Emily C. Zabor [aut] (<https://orcid.org/0000-0002-1402-4498>),
Michael Curry [ctb] (<https://orcid.org/0000-0002-0261-4044>),
Esther Drill [ctb] (<https://orcid.org/0000-0002-3315-4538>),
Jessica Flynn [ctb] (<https://orcid.org/0000-0001-8310-6684>)

Maintainer Daniel D. Sjoberg <danield.sjoberg@gmail.com>

Repository CRAN

Date/Publication 2019-08-20 21:20:05 UTC

R topics documented:

`add_global_p` *Adds the global p-value for a categorical variables*

Description

This function uses car::Anova with argument `type = "III"` to calculate global p-values for categorical variables. Output from `tbl_regression` and `tbl_uvregression` objects supported.

Usage

```
add_global_p(x, ...)
```

Arguments

`x`	`tbl_regression` or `tbl_uvregression` object
`...`	Further arguments passed to or from other methods.

Note

If a needed class of model is not supported by car::Anova, please create an issue to request support.

Author(s)

Daniel D. Sjoberg

See Also

`add_global_p.tbl_regression`, `add_global_p.tbl_uvregression`

`add_global_p.tbl_regression` *Adds the global p-value for categorical variables*

Description

This function uses car::Anova with argument `type = "III"` to calculate global p-values for categorical variables.

Usage

```
## S3 method for class 'tbl_regression'
add_global_p(x, terms = NULL, keep = FALSE,
  ...)
```

Arguments

`x`	Object with class `tbl_regression` from the tbl_regression function
`terms`	Character vector of terms for which to add global p-values. Default is `NULL` which will add global p-values for all categorical variables
`keep`	Logical argument indicating whether to also retain the individual p-values in the table output for each level of the categorical variable. Default is `FALSE`
`...`	Additional arguments to be passed to car::Anova

Value

A `tbl_regression` object

Note

If a needed class of model is not supported by car::Anova, please create an issue to request support.

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_regression tools: `add_nevent.tbl_regression`, `bold_italicize_labels_levels`, `bold_p.tbl_regression`, `bold_p.tbl_stack`, `inline_text.tbl_regression`, `modify_header`, `sort_p.tbl_regression`, `tbl_merge`, `tbl_regression`, `tbl_stack`

Examples

```
tbl_lm_global_ex1 <-
  lm(marker ~ age + grade, trial) %>%
  tbl_regression() %>%
  add_global_p()
```

`add_global_p.tbl_uvregression` *Adds the global p-value for categorical variables*

Description

This function uses car::Anova with argument `type = "III"` to calculate global p-values for categorical variables.

Usage

```
## S3 method for class 'tbl_uvregression'
add_global_p(x, ...)
```

Arguments

`x`	Object with class `tbl_uvregression` from the tbl_uvregression function
`...`	Additional arguments to be passed to car::Anova.

Value

A `tbl_uvregression` object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_uvregression tools: `add_nevent.tbl_uvregression`, `add_q.tbl_uvregression`, `bold_italicize_labels_lev`
`bold_p.tbl_stack`, `bold_p.tbl_uvregression`, `inline_text.tbl_uvregression`, `modify_header`,
`sort_p.tbl_uvregression`, `tbl_merge`, `tbl_stack`, `tbl_uvregression`

Examples

```
tbl_uv_global_ex2 <-
  trial %>%
  dplyr::select(response, trt, age, grade) %>%
  tbl_uvregression(
    method = glm,
    y = response,
    method.args = list(family = binomial),
    exponentiate = TRUE
  ) %>%
  add_global_p()
```

---

add_n *Add column with N*

Description

For each variable in a `tbl_summary` table, the `add_n` function adds a column with the total number of non-missing (or missing) observations

Usage

```
add_n(x, missing = FALSE, last = FALSE)
```

Arguments

`x`	Object with class `tbl_summary` from the tbl_summary function
`missing`	Logical argument indicating whether to print N (`missing = FALSE`), or N missing (`missing = TRUE`). Default is `FALSE`
`last`	Logical indicator to include N column last in table. Default is `FALSE`, which will display N column first.

Value

A `tbl_summary` object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_summary tools: `add_overall`, `add_p`, `add_q.tbl_summary`, `add_stat_label`, `bold_italicize_labels_levels`, `bold_p.tbl_summary`, `inline_text.tbl_summary`, `modify_header`, `sort_p.tbl_summary`, `tbl_summary`

Examples

```
tbl_n_ex <-
  trial %>%
  dplyr::select(trt, age, grade, response) %>%
  tbl_summary(by = trt) %>%
  add_n()
```

`add_nevent` *Add number of events to a regression table*

Description

Adds a column of the number of events to tables created with tbl_regression or tbl_uvregression. Supported model types include GLMs with binomial distribution family (e.g. stats::glm, lme4::glmer, and geepack::geeglm) and Cox Proportion Hazards regression models (survival::coxph).

Usage

```
add_nevent(x, ...)
```

Arguments

`x`	`tbl_regerssion` or `tbl_uvregression` object
`...`	Additional arguments passed to or from other methods.

Author(s)

Daniel D. Sjoberg

See Also

add_nevent.tbl_regression, add_nevent.tbl_uvregression, tbl_regression, tbl_uvregression

---

`add_nevent.tbl_regression` *Add number of events to a regression table*

Description

This function adds a column of the number of events to tables created with tbl_regression. Supported model types include GLMs with binomial distribution family (e.g. stats::glm, lme4::glmer, and geepack::geeglm) and Cox Proportion Hazards regression models (survival::coxph).

Usage

```
## S3 method for class 'tbl_regression'
add_nevent(x, ...)
```

Arguments

x	tbl_regression object
...	Not used

Value

A tbl_regression object

Reporting Event N

The number of events is added to the internal .$table_body tibble, and not printed in the default output table (similar to N). The number of events is accessible via the inline_text function for printing in a report.

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_regression tools: add_global_p.tbl_regression, bold_italicize_labels_levels, bold_p.tbl_regression, bold_p.tbl_stack, inline_text.tbl_regression, modify_header, sort_p.tbl_regression, tbl_merge, tbl_regression, tbl_stack

Examples

```
tbl_reg_nevent_ex <-
  glm(response ~ trt, trial, family = binomial) %>%
  tbl_regression() %>%
  add_nevent()
```

---

add_nevent.tbl_uvregression

Add number of events to a regression table

Description

Adds a column of the number of events to tables created with tbl_uvregression. Supported model types include GLMs with binomial distribution family (e.g. stats::glm, lme4::glmer, and geepack::geeglm) and Cox Proportion Hazards regression models (survival::coxph).

Usage

```
## S3 method for class 'tbl_uvregression'
add_nevent(x, ...)
```

Arguments

x	tbl_uvregerssion object
...	Not used

Value

A tbl_uvregression object

Reporting Event N

The number of events is added to the internal .$table_body tibble, and printed to the right of the N column. The number of events is also accessible via the inline_text function for printing in a report.

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_q.tbl_uvregression, bold_italicize_labels_l
bold_p.tbl_stack, bold_p.tbl_uvregression, inline_text.tbl_uvregression, modify_header,
sort_p.tbl_uvregression, tbl_merge, tbl_stack, tbl_uvregression

Examples

```
tbl_uv_nevent_ex <-
  trial %>%
  dplyr::select(response, trt, age, grade) %>%
  tbl_uvregression(
    method = glm,
    y = response,
    method.args = list(family = binomial)
  ) %>%
  add_nevent()
```

---

add_overall *Add column with overall summary statistics*

Description

Adds a column with overall summary statistics to tables created by tbl_summary.

Usage

```
add_overall(x, last = FALSE)
```

Arguments

x	Object with class tbl_summary from the tbl_summary function
last	Logical indicator to display overall column last in table. Default is FALSE, which will display overall column first.

Value

A tbl_summary object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_summary tools: add_n, add_p, add_q.tbl_summary, add_stat_label, bold_italicize_labels_levels, bold_p.tbl_summary, inline_text.tbl_summary, modify_header, sort_p.tbl_summary, tbl_summary

Examples

```
tbl_overall_ex <-
  trial %>%
  dplyr::select(age, response, grade, trt) %>%
  tbl_summary(by = trt) %>%
  add_overall()
```

---

add_p *Adds p-values to summary tables*

Description

Adds p-values to tables created by tbl_summary by comparing values across groups.

Usage

```
add_p(x, test = NULL, pvalue_fun = NULL, group = NULL,
  include = NULL, exclude = NULL)
```

Arguments

x	Object with class tbl_summary from the tbl_summary function
test	List of formulas specifying statistical tests to perform, e.g. list(all_continuous() ~ "t.test",all_categorical() ~ "fisher.test"). Options include • "t.test" for a t-test, • "wilcox.test" for a Wilcoxon rank-sum test, • "kruskal.test" for a Kruskal-Wallis rank-sum test, • "chisq.test" for a Chi-squared test of independence, • "fisher.test" for a Fisher's exact test, • "lme4" for a random intercept logistic regression model to account for clustered data, lme4::glmer(by ~ variable + (1 | group),family = binomial). The by argument must be binary for this option. Tests default to "kruskal.test" for continuous variables, "chisq.test" for categorical variables with all expected cell counts >=5, and "fisher.test" for categorical variables with any expected cell count <5. A custom test function can be added for all or some variables. See below for an example.
pvalue_fun	Function to round and format p-values. Default is style_pvalue. The function must have a numeric vector input (the numeric, exact p-value), and return a string that is the rounded/formatted p-value (e.g. pvalue_fun = function(x) style_pvalue(x,digits = 2) or equivalently, purrr::partial(style_pvalue,digits = 2)).
group	Column name of an ID or grouping variable. The column can be used calculate p-values with correlated data (e.g. when the test argument is "lme4"). Default is NULL. If specified, the row associated with this variable is omitted from the summary table.
include	Names of variables to include in output.
exclude	Names of variables to exclude from output.

Value

A tbl_summary object

Setting Defaults

If you like to consistently use a different function to format p-values or estimates, you can set options in the script or in the user- or project-level startup file, '.Rprofile'. The default confidence level can also be set. Please note the default option for the estimate is the same as it is for tbl_regression().

- options(gtsummary.pvalue_fun = new_function)

Example Output

Author(s)

Emily C. Zabor, Daniel D. Sjoberg

See Also

See tbl_summary vignette for detailed examples

Other tbl_summary tools: add_n, add_overall, add_q.tbl_summary, add_stat_label, bold_italicize_labels_levels
bold_p.tbl_summary, inline_text.tbl_summary, modify_header, sort_p.tbl_summary, tbl_summary

Examples

```
add_p_ex1 <-
  trial %>%
  dplyr::select(age, grade, response, trt) %>%
  tbl_summary(by = trt) %>%
  add_p()


# Conduct a custom McNemar test for response,
# Function must return a named list(p = 0.05, test = "McNemar's test")
# Function names begins with 'add_p_test.' and ends with the alias
add_p_test.mcnemar <- function(data, variable, by, ...) {
  result <- list()
  result$p <- stats::mcnemar.test(data[[variable]], data[[by]])$p.value
  result$test <- "McNemar's test"
  result
}

add_p_ex2 <-
  trial[c("response", "trt")] %>%
  tbl_summary(by = trt) %>%
  add_p(test = vars(response) ~ "mcnemar")
```

---

add_q *Add a column of q values to account for multiple comparisons*

Description

Add a column of q values to account for multiple comparisons

Usage

```
add_q(x, ...)
```

Arguments

x	tbl_summary or tbl_uvregression object
...	Additional arguments passed to other methods.

Author(s)

Esther Drill, Daniel D. Sjoberg

See Also

add_q.tbl_summary, add_q.tbl_uvregression, tbl_summary, tbl_uvregression

---

add_q.tbl_summary *Add a column of q-values to account for multiple comparisons*

Description

Adjustments to are p-values are performed with stats::p.adjust.

Usage

```
## S3 method for class 'tbl_summary'
add_q(x, method = "fdr",
  pvalue_fun = x$pvalue_fun, ...)
```

Arguments

x	tbl_summary object
method	String indicating method to be used for p-value adjustment. Methods from stats::p.adjust are accepted. Default is method = 'fdr'.
pvalue_fun	Function to round and format p-values. Default is style_pvalue. The function must have a numeric vector input (the numeric, exact p-value), and return a string that is the rounded/formatted p-value (e.g. pvalue_fun = function(x) style_pvalue(x,digits = 2) or equivalently, purrr::partial(style_pvalue,digits = 2)).
...	Additional arguments passed to or from other methods

Value

A tbl_summary object

Example Output

Author(s)

Esther Drill, Daniel D. Sjoberg

See Also

Other tbl_summary tools: add_n, add_overall, add_p, add_stat_label, bold_italicize_labels_levels, bold_p.tbl_summary, inline_text.tbl_summary, modify_header, sort_p.tbl_summary, tbl_summary

Examples

```
tbl_sum_q_ex <-
  trial %>%
  dplyr::select(trt, age, grade, response) %>%
  tbl_summary(by = trt) %>%
  add_p() %>%
  add_q()
```

`add_q.tbl_uvregression` *Add a column of q-values to account for multiple comparisons*

Description

Adjustments to are p-values are performed with stats::p.adjust.

Usage

```
## S3 method for class 'tbl_uvregression'
add_q(x, method = "fdr",
  pvalue_fun = x$inputs$pvalue_fun, ...)
```

Arguments

`x`	`tbl_uvregression` object
`method`	String indicating method to be used for p-value adjustment. Methods from stats::p.adjust are accepted. Default is `method = 'fdr'`.
`pvalue_fun`	Function to round and format p-values. Default is style_pvalue. The function must have a numeric vector input (the numeric, exact p-value), and return a string that is the rounded/formatted p-value (e.g. `pvalue_fun = function(x) style_pvalue(x,digits = 2)` or equivalently, `purrr::partial(style_pvalue,digits = 2)`).
`...`	Additional arguments passed to or from other methods

Value

A `tbl_uvregression` object

Example Output

Author(s)

Esther Drill, Daniel D. Sjoberg

See Also

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_nevent.tbl_uvregression, bold_italicize_labels_levels, bold_p.tbl_stack, bold_p.tbl_uvregression, inline_text.tbl_uvregression, modify_header, sort_p.tbl_uvregression, tbl_merge, tbl_stack, tbl_uvregression

Examples

```
tbl_uvr_q_ex <-
  trial %>%
  dplyr::select(age, marker, grade, response) %>%
  tbl_uvregression(
    method = lm,
    y = age
  ) %>%
  add_global_p() %>%
  add_q()
```

---

add_stat_label *Add statistic labels column*

Description

Adds a column with labels describing the summary statistics presented for each variable in the tbl_summary table.

Usage

```
add_stat_label(x)
```

Arguments

x	Object with class tbl_summary from the tbl_summary function

Value

A tbl_summary object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_summary tools: add_n, add_overall, add_p, add_q.tbl_summary, bold_italicize_labels_levels, bold_p.tbl_summary, inline_text.tbl_summary, modify_header, sort_p.tbl_summary, tbl_summary

Examples

```
tbl_stat_ex <-
  trial %>%
  dplyr::select(trt, age, grade, response) %>%
  tbl_summary() %>%
  add_stat_label()
```

---

as_gt *Convert gtsummary object to a gt_tbl object*

Description

Function converts gtsummary objects to a gt_tbl objects. Function is used in the background when the results are printed or knit. A user can use this function if they wish to add customized formatting available via the gt package. Review the tbl_summary vignette or tbl_regression vignette for detailed examples in the 'Advanced Customization' section.

Usage

```
as_gt(x, include = NULL, exclude = NULL, omit = NULL)
```

Arguments

x	Object created by a function from the gtsummary package (e.g. tbl_summary or tbl_regression)
include	Character vector naming gt commands to include in printing. Default is NULL, which utilizes all commands in x$gt_calls.
exclude	Character vector naming gt commands to exclude in printing. Default is NULL.
omit	DEPRECATED. Argument is synonymous with exclude vector of named gt commands to omit. Default is NULL

Value

A gt_tbl object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

tbl_summary tbl_regression tbl_uvregression tbl_survival

Examples

```
as_gt_ex <-
  trial[c("trt", "age", "response", "grade")] %>%
  tbl_summary(by = trt) %>%
  as_gt()
```

`as_kable` *Convert to knitr_kable object*

Description

Function converts gtsummary objects to a knitr_kable objects. This function is used in the background when the results are printed or knit. A user can use this function if they wish to add customized formatting available via knitr::kable.

Usage

```
as_kable(x, include = NULL, exclude = NULL, ...)
```

Arguments

`x`	Object created by a function from the gtsummary package (e.g. tbl_summary or tbl_regression)
`include`	Character vector naming kable commands to include in printing. Default is `NULL`, which utilizes all commands in `x$kable_calls`.
`exclude`	Character vector naming kable commands to exclude in printing. Default is `NULL`.
`...`	Additional arguments passed to knitr::kable

Value

A `knitr_kable` object

Author(s)

Daniel D. Sjoberg

See Also

tbl_summary tbl_regression tbl_uvregression tbl_survival

Examples

```
trial %>%
  tbl_summary(by = trt) %>%
  as_kable()
```

`bold_italicize_labels_levels` *Bold or Italicize labels or levels in gtsummary tables*

Description

Bold or Italicize labels or levels in gtsummary tables

Usage

```
bold_labels(x)

bold_levels(x)

italicize_labels(x)

italicize_levels(x)
```

Arguments

`x`	Object created using gtsummary functions

Value

Functions return the same class of gtsummary object supplied

Functions

- `bold_labels`: Bold labels in gtsummary tables
- `bold_levels`: Bold levels in gtsummary tables
- `italicize_labels`: Italicize labels in gtsummary tables
- `italicize_levels`: Italicize levels in gtsummary tables

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_summary tools: add_n, add_overall, add_p, add_q.tbl_summary, add_stat_label, bold_p.tbl_summary, inline_text.tbl_summary, modify_header, sort_p.tbl_summary, tbl_summary

Other tbl_regression tools: add_global_p.tbl_regression, add_nevent.tbl_regression, bold_p.tbl_regression, bold_p.tbl_stack, inline_text.tbl_regression, modify_header, sort_p.tbl_regression, tbl_merge, tbl_regression, tbl_stack

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_nevent.tbl_uvregression, add_q.tbl_uvregression, bold_p.tbl_stack, bold_p.tbl_uvregression, inline_text.tbl_uvregression, modify_header, sort_p.tbl_uvregression, tbl_merge, tbl_stack, tbl_uvregression

Examples

```
tbl_bold_ital_ex <-
  trial %>%
  dplyr::select(trt, age, grade) %>%
  tbl_summary() %>%
  bold_labels() %>%
  bold_levels() %>%
  italicize_labels() %>%
  italicize_levels()
```

---

bold_p *Bold significant p-values or q-values*

Description

Bold values below a chosen threshold (e.g. <0.05) in gtsummary tables.

Usage

```
bold_p(x, ...)
```

Arguments

x	Object created using gtsummary functions
...	Additional arguments passed to other methods.

Author(s)

Daniel D. Sjoberg, Esther Drill

See Also

bold_p.tbl_summary, bold_p.tbl_regression, bold_p.tbl_uvregression

`bold_p.tbl_regression` *Bold significant p-values or q-values*

Description

Bold values below a chosen threshold (e.g. <0.05) in tbl_regression tables.

Usage

```
## S3 method for class 'tbl_regression'
bold_p(x, t = 0.05, ...)
```

Arguments

`x`	Object created using tbl_regression function
`t`	Threshold below which values will be bold. Default is 0.05.
`...`	Not used

Value

A `tbl_regression` object

Example Output

Author(s)

Daniel D. Sjoberg, Esther Drill

See Also

Other tbl_regression tools: `add_global_p.tbl_regression`, `add_nevent.tbl_regression`, `bold_italicize_labels_le`
`bold_p.tbl_stack`, `inline_text.tbl_regression`, `modify_header`, `sort_p.tbl_regression`,
`tbl_merge`, `tbl_regression`, `tbl_stack`

Examples

```
tbl_lm_bold_p_ex <-
  glm(response ~ trt + grade, trial, family = binomial(link = "logit")) %>%
  tbl_regression(exponentiate = TRUE) %>%
  bold_p()
```

`bold_p.tbl_stack` *Bold significant p-values or q-values*

Description

Bold values below a chosen threshold (e.g. <0.05) in tbl_stack tables.

Usage

```
## S3 method for class 'tbl_stack'
bold_p(x, ...)
```

Arguments

`x`	Object created using tbl_stack function
`...`	arguments passed to `bold_p.*()` method that matches the first object in the `tbl_stack`

Value

A `tbl_stack` object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_uvregression tools: `add_global_p.tbl_uvregression`, `add_nevent.tbl_uvregression`, `add_q.tbl_uvregression`, `bold_italicize_labels_levels`, `bold_p.tbl_uvregression`, `inline_text.tbl_uvregres` `modify_header`, `sort_p.tbl_uvregression`, `tbl_merge`, `tbl_stack`, `tbl_uvregression`

Other tbl_regression tools: `add_global_p.tbl_regression`, `add_nevent.tbl_regression`, `bold_italicize_labels_le` `bold_p.tbl_regression`, `inline_text.tbl_regression`, `modify_header`, `sort_p.tbl_regression`, `tbl_merge`, `tbl_regression`, `tbl_stack`

Examples

```
t1 <- tbl_regression(lm(age ~ response, trial))
t2 <- tbl_regression(lm(age ~ grade, trial))

bold_p_stack_ex <-
  tbl_stack(list(t1, t2)) %>%
  bold_p(t = 0.10)
```

`bold_p.tbl_summary` *Bold significant p-values or q-values*

Description

Bold values below a chosen threshold (e.g. <0.05) in tbl_summary tables.

Usage

```
## S3 method for class 'tbl_summary'
bold_p(x, t = 0.05, q = FALSE, ...)
```

Arguments

`x`	Object created using `tbl_summary` function
`t`	Threshold below which values will be bold. Default is 0.05.
`q`	Logical argument. When TRUE will bold the q-value column rather than the p-values. Default is `FALSE`.
`...`	Not used

Value

A `tbl_summary` object

Example Output

Author(s)

Daniel D. Sjoberg, Esther Drill

See Also

Other tbl_summary tools: `add_n`, `add_overall`, `add_p`, `add_q.tbl_summary`, `add_stat_label`, `bold_italicize_labels_levels`, `inline_text.tbl_summary`, `modify_header`, `sort_p.tbl_summary`, `tbl_summary`

Examples

```
tbl_sum_bold_p_ex <-
  trial %>%
  dplyr::select(age, grade, response, trt) %>%
  tbl_summary(by = trt) %>%
  add_p() %>%
  bold_p()
```

`bold_p.tbl_uvregression` *Bold significant p-values or q-values*

Description

Bold values below a chosen threshold (e.g. <0.05) in tbl_uvregression tables.

Usage

```
## S3 method for class 'tbl_uvregression'
bold_p(x, t = 0.05, q = FALSE, ...)
```

Arguments

`x`	Object created using tbl_uvregression function
`t`	Threshold below which values will be bold. Default is 0.05.
`q`	Logical argument. When TRUE will bold the q-value column rather than the p-values. Default is `FALSE`.
`...`	Not used

Value

A `tbl_uvregression` object

Example Output

Author(s)

Daniel D. Sjoberg, Esther Drill

See Also

Other tbl_uvregression tools: `add_global_p.tbl_uvregression`, `add_nevent.tbl_uvregression`, `add_q.tbl_uvregression`, `bold_italicize_labels_levels`, `bold_p.tbl_stack`, `inline_text.tbl_uvregression`, `modify_header`, `sort_p.tbl_uvregression`, `tbl_merge`, `tbl_stack`, `tbl_uvregression`

Examples

```
tbl_uvglm_bold_p_ex <-
  trial %>%
  dplyr::select(age, marker, response, grade) %>%
  tbl_uvregression(
    method = glm,
    y = response,
    method.args = list(family = binomial),
```

```
  exponentiate = TRUE
) %>%
bold_p(t = 0.25)
```

`gtsummary_logo` *The gtsummary logo, using ASCII or Unicode characters*

Description

Use `crayon::strip_style()` to get rid of the colors.

Usage

```
gtsummary_logo(unicode = l10n_info()$`UTF-8`)
```

Arguments

`unicode`	Whether to use Unicode symbols. Default is `TRUE` on UTF-8 platforms.

Examples

```
gtsummary_logo()
```

`inline_text` *Report statistics from gtsummary tables inline*

Description

Report statistics from gtsummary tables inline

Usage

```
inline_text(x, ...)
```

Arguments

`x`	Object created from a gtsummary function
`...`	Additional arguments passed to other methods.

Value

A string reporting results from a gtsummary table

Author(s)

Daniel D. Sjoberg

See Also

inline_text.tbl_summary, inline_text.tbl_regression, inline_text.tbl_uvregression, inline_text.tbl_survival

---

`inline_text.tbl_regression`

Report statistics from regression summary tables inline

Description

Takes an object with class `tbl_regression`, and the location of the statistic to report and returns statistics for reporting inline in an R markdown document. Detailed examples in the tbl_regression vignette

Usage

```
## S3 method for class 'tbl_regression'
inline_text(x, variable, level = NULL,
 pattern = "{estimate} ({conf.level*100}% CI {conf.low}, {conf.high}; {p.value})",
  estimate_fun = x$inputs$estimate_fun, pvalue_fun = function(x)
  style_pvalue(x, prepend_p = TRUE), ...)
```

Arguments

`x`	Object created from tbl_regression
`variable`	Variable name of statistics to present
`level`	Level of the variable to display for categorical variables. Default is `NULL`, returning the top row in the table for the variable.
`pattern`	String indicating the statistics to return. Uses glue::glue formatting. Default is `"{estimate} ({conf.level }% CI {conf.low},{conf.high}; {p.value})"`. All columns from `x$table_body` are available to print as well as the confidence level (conf.level). See below for details.
`estimate_fun`	function to style model coefficient estimates. Columns 'estimate', 'conf.low', and 'conf.high' are formatted. Default is `x$inputs$estimate_fun`
`pvalue_fun`	function to style p-values and/or q-values. Default is `function(x) style_pvalue(x,prepend_p = TRUE)`
`...`	Not used

Value

A string reporting results from a gtsummary table

pattern argument

The following items are available to print. Use `print(x$table_body)` to print the table the estimates are extracted from.

- `{estimate}` coefficient estimate formatted with 'estimate_fun'
- `{conf.low}` lower limit of confidence interval formatted with 'estimate_fun'
- `{conf.high}` upper limit of confidence interval formatted with 'estimate_fun'
- `{ci}` confidence interval formatted with x$estimate_fun
- `{p.value}` p-value formatted with 'pvalue_fun'
- `{N}` number of observations in model
- `{label}` variable/variable level label

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_regression tools: add_global_p.tbl_regression, add_nevent.tbl_regression, bold_italicize_labels_le
bold_p.tbl_regression, bold_p.tbl_stack, modify_header, sort_p.tbl_regression, tbl_merge,
tbl_regression, tbl_stack

Examples

```
inline_text_ex1 <-
  glm(response ~ age + grade, trial, family = binomial(link = "logit")) %>%
  tbl_regression(exponentiate = TRUE)

inline_text(inline_text_ex1, variable = "age")
inline_text(inline_text_ex1, variable = "grade", level = "III")
```

---

inline_text.tbl_summary

Report statistics from summary tables inline

Description

Extracts and returns statistics from a `tbl_summary` object for inline reporting in an R markdown document. Detailed examples in the tbl_summary vignette

Usage

```
## S3 method for class 'tbl_summary'
inline_text(x, variable, level = NULL,
  column = ifelse(is.null(x$by), "stat_0", stop("Must specify column")),
  pvalue_fun = function(x) style_pvalue(x, prepend_p = TRUE), ...)
```

Arguments

x	Object created from tbl_summary
variable	Variable name of statistic to present
level	Level of the variable to display for categorical variables. Can also specify the 'Unknown' row. Default is NULL
column	Column name to return from x$table_body. Can also pass the level of a by variable.
pvalue_fun	Function to round and format p-values. Default is style_pvalue. The function must have a numeric vector input (the numeric, exact p-value), and return a string that is the rounded/formatted p-value (e.g. pvalue_fun = function(x) style_pvalue(x,digits = 2) or equivalently, purrr::partial(style_pvalue,digits = 2)).
...	Not used

Value

A string reporting results from a gtsummary table

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_summary tools: add_n, add_overall, add_p, add_q.tbl_summary, add_stat_label, bold_italicize_labels_levels, bold_p.tbl_summary, modify_header, sort_p.tbl_summary, tbl_summary

Examples

```
t1 <- tbl_summary(trial)
t2 <- tbl_summary(trial, by = trt) %>% add_p()

inline_text(t1, variable = "age")
inline_text(t2, variable = "grade", level = "I", column = "Drug")
inline_text(t2, variable = "grade", column = "p.value")
```

---

inline_text.tbl_survival

Report statistics from survival summary tables inline

Description

for inline reporting in an R markdown document.

Usage

```
## S3 method for class 'tbl_survival'
inline_text(x, strata = NULL, time = NULL,
  prob = NULL, pattern = "{estimate} ({conf.level*100}% CI {ci})",
  estimate_fun = x$estimate_fun, ...)
```

Arguments

x	Object created from tbl_survival
strata	If tbl_survival estimates are stratified, level of the stratum to report. Default is NULL when tbl_survival have no specified strata.
time	Time for which to return survival probability
prob	Probability for which to return survival time. For median survival use prob = 0.50
pattern	String indicating the statistics to return. Uses glue::glue formatting. Default is '{estimate} ({conf.level*100}% {ci})'. All columns from x$table_long are available to print as well as the confidence level (conf.level). See below for details.
estimate_fun	function to round/style estimate and lower/upper confidence interval estimates. Note, this does not style the 'ci' column, which is a string. Default is x$estimate_fun
...	Not used

Value

A string reporting results from a gtsummary table

pattern argument

The following items are available to print. Use print(x$table_long) to print the table the estimates are extracted from.

- {label} 'time' or 'prob' label
- {estimate} survival or survival time estimate formatted with 'estimate_fun'
- {conf.low} lower limit of confidence interval formatted with 'estimate_fun'
- {conf.high} upper limit of confidence interval formatted with 'estimate_fun'
- {ci} confidence interval formatted with x$estimate_fun (pre-formatted)
- {time}/{prob} time or survival quantile (numeric)
- {n.risk} number at risk at 'time' (within stratum if applicable)
- {n.event} number of observed events at 'time' (within stratum if applicable)
- {n} number of observations (within stratum if applicable)
- {variable} stratum variable (if applicable)
- {level} stratum level (if applicable)
- {groupname} label_level from original tbl_survival() call

Author(s)

Karissa Whiting

See Also

Other tbl_survival tools: modify_header, tbl_survival.survfit

Examples

```
library(survival)
surv_table <-
  survfit(Surv(ttdeath, death) ~ trt, trial) %>%
  tbl_survival(times = c(12, 24))

inline_text(surv_table,
  strata = "Drug",
  time = 12
)
```

---

inline_text.tbl_uvregression

Report statistics from regression summary tables inline

Description

Extracts and returns statistics from a table created by the tbl_uvregression function for inline reporting in an R markdown document. Detailed examples in the tbl_regression vignette

Usage

```
## S3 method for class 'tbl_uvregression'
inline_text(x, variable, level = NULL,
 pattern = "{estimate} ({conf.level*100}% CI {conf.low}, {conf.high}; {p.value})",
  estimate_fun = x$inputs$estimate_fun, pvalue_fun = function(x)
  style_pvalue(x, prepend_p = TRUE), ...)
```

Arguments

x	Object created from tbl_uvregression
variable	Variable name of statistics to present
level	Level of the variable to display for categorical variables. Default is NULL, returning the top row in the table for the variable.
pattern	String indicating the statistics to return. Uses glue::glue formatting. Default is "{estimate} ({conf.level }% CI {conf.low},{conf.high}; {p.value})". All columns from x$table_body are available to print as well as the confidence level (conf.level). See below for details.

estimate_fun	function to style model coefficient estimates. Columns 'estimate', 'conf.low', and 'conf.high' are formatted. Default is `x$inputs$estimate_fun`
pvalue_fun	function to style p-values and/or q-values. Default is `function(x) style_pvalue(x,prepend_p = TRUE)`
...	Not used

Value

A string reporting results from a gtsummary table

pattern argument

The following items are available to print. Use `print(x$table_body)` to print the table the estimates are extracted from.

- {estimate} coefficient estimate formatted with 'estimate_fun'
- {conf.low} lower limit of confidence interval formatted with 'estimate_fun'
- {conf.high} upper limit of confidence interval formatted with 'estimate_fun'
- {ci} confidence interval formatted with x$estimate_fun
- {p.value} p-value formatted with 'pvalue_fun'
- {N} number of observations in model
- {label} variable/variable level label

See Also

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_nevent.tbl_uvregression, add_q.tbl_uvregression, bold_italicize_labels_levels, bold_p.tbl_stack, bold_p.tbl_uvregression, modify_header, sort_p.tbl_uvregression, tbl_merge, tbl_stack, tbl_uvregression

Examples

```
inline_text_ex1 <-
  trial %>%
  dplyr::select(response, age, grade) %>%
  tbl_uvregression(
    method = glm,
    method.args = list(family = binomial),
    y = response,
    exponentiate = TRUE
  )

inline_text(inline_text_ex1, variable = "age")
inline_text(inline_text_ex1, variable = "grade", level = "III")
```

`modify_header` *Modify column headers in gtsummary tables*

Description

Column labels can be modified to include calculated statistics; e.g. the N can be dynamically included by wrapping it in curly brackets (following glue::glue syntax).

Usage

```
modify_header(x, stat_by = NULL, ..., text_interpret = c("md", "html"))
```

Arguments

`x`	gtsummary object, e.g. `tbl_summary` or `tbl_regression`
`stat_by`	String specifying text to include above the summary statistics stratified by a variable. Only use with stratified `tbl_summary` objects. The following fields are available for use in the headers: • `{n}` number of observations in each group, • `{N}` total number of observations, • `{p}` percentage in each group, • `{level}` the 'by' variable level, • `"fisher.test"` for a Fisher's exact test, Syntax follows glue::glue, e.g. `stat_by = "**{level}**, N = {n} ({style_percent(p)\%})"`. The by argument from the parent `tbl_summary()` cannot be `NULL`.
`...`	Specifies column label of any other column in `.$table_body`. Argument is the column name, and the value is the new column header (e.g. `p.value = "Model P-values"`). Use `print(x$table_body)` to see columns available.
`text_interpret`	indicates whether text will be interpreted as markdown (`"md"`) or HTML (`"html"`). The text is interpreted with the gt package's `md()` or `html()` functions. The default is `"md"`, and is ignored when the print engine is not gt.

Value

Function return the same class of gtsummary object supplied

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_summary tools: add_n, add_overall, add_p, add_q.tbl_summary, add_stat_label, bold_italicize_labels_levels, bold_p.tbl_summary, inline_text.tbl_summary, sort_p.tbl_summary, tbl_summary

Other tbl_regression tools: add_global_p.tbl_regression, add_nevent.tbl_regression, bold_italicize_labels_le bold_p.tbl_regression, bold_p.tbl_stack, inline_text.tbl_regression, sort_p.tbl_regression, tbl_merge, tbl_regression, tbl_stack

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_nevent.tbl_uvregression, add_q.tbl_uvregression, bold_italicize_labels_levels, bold_p.tbl_stack, bold_p.tbl_uvregression, inline_text.tbl_uvregression, sort_p.tbl_uvregression, tbl_merge, tbl_stack, tbl_uvregression

Other tbl_survival tools: inline_text.tbl_survival, tbl_survival.survfit

Examples

```
tbl_col_ex1 <-
  trial[c("age", "grade", "response")] %>%
  tbl_summary() %>%
  modify_header(stat_0 = "**All Patients**, N = {N}")

tbl_col_ex2 <-
  trial[c("age", "grade", "response", "trt")] %>%
  tbl_summary(by = trt) %>%
  modify_header(
    stat_by = "**{level}**, N = {n} ({style_percent(p, symbol = TRUE)})"
  )
```

---

print_gtsummary *print and knit_print methods for gtsummary objects*

Description

print and knit_print methods for gtsummary objects

Usage

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

## S3 method for class 'tbl_summary'
knit_print(x, ...)

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

## S3 method for class 'tbl_regression'
knit_print(x, ...)
```

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

## S3 method for class 'tbl_uvregression'
knit_print(x, ...)

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

## S3 method for class 'tbl_survival'
knit_print(x, ...)

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

## S3 method for class 'tbl_merge'
knit_print(x, ...)

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

## S3 method for class 'tbl_stack'
knit_print(x, ...)
```

Arguments

x	An object created using gtsummary functions
...	Not used

Author(s)

Daniel D. Sjoberg

See Also

tbl_summary tbl_regression tbl_uvregression

`select_helpers` *Select helper functions*

Description

Set of functions to supplement the tidyselect set of functions for selecting columns of data frames. `all_continuous()`, `all_categorical()`, and `all_dichotomous()` may only be used with `tbl_summary()`, where each variable has been classified into one of these three groups. All other helpers are available throughout the package.

Usage

```
all_numeric()

all_character()

all_integer()

all_double()

all_logical()

all_factor()

all_continuous()

all_categorical(dichotomous = TRUE)

all_dichotomous()
```

Arguments

dichotomous	Logical indicating whether to include dichotomous variables. Default is TRUE

Value

A character vector of column names selected

`sort_p.tbl_regression` *Sort variables in table by ascending p-values*

Description

Sort variables in tables created by tbl_regression by ascending p-values

Usage

```
## S3 method for class 'tbl_regression'
sort_p(x, ...)
```

Arguments

x	An object created using `tbl_regression` function
...	Not used

Value

A `tbl_regression` object

Example Output

Author(s)

Karissa Whiting

See Also

Other tbl_regression tools: add_global_p.tbl_regression, add_nevent.tbl_regression, bold_italicize_labels_le
bold_p.tbl_regression, bold_p.tbl_stack, inline_text.tbl_regression, modify_header,
tbl_merge, tbl_regression, tbl_stack

Examples

```
tbl_lm_sort_p_ex <-
  glm(response ~ trt + grade, trial, family = binomial(link = "logit")) %>%
  tbl_regression(exponentiate = TRUE) %>%
  sort_p()
```

---

sort_p.tbl_summary *Sort variables in table by ascending p-values*

Description

Sort variables in tables created by tbl_summary by ascending p-values

Usage

```
## S3 method for class 'tbl_summary'
sort_p(x, q = FALSE, ...)
```

Arguments

x	An object created using tbl_summary function
q	Logical argument. When TRUE will sort by the q-value column rather than the p-values
...	Not used

Value

A tbl_summary object

Example Output

Author(s)

Karissa Whiting

See Also

Other tbl_summary tools: add_n, add_overall, add_p, add_q.tbl_summary, add_stat_label, bold_italicize_labels_levels, bold_p.tbl_summary, inline_text.tbl_summary, modify_header, tbl_summary

Examples

```
tbl_sum_sort_p_ex <-
  trial %>%
  dplyr::select(age, grade, response, trt) %>%
  tbl_summary(by = trt) %>%
  add_p() %>%
  sort_p()
```

---

sort_p.tbl_uvregression

Sort variables in table by ascending p-values

Description

Sort variables in tables created by tbl_uvregression by ascending p-values

Usage

```
## S3 method for class 'tbl_uvregression'
sort_p(x, q = FALSE, ...)
```

Arguments

x	an object created using tbl_uvregression function
q	logical argument. When TRUE will sort by the q-value column rather than the p-values
...	Not used

Value

A tbl_uvregression object

Example Output

Author(s)

Karissa Whiting

See Also

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_nevent.tbl_uvregression, add_q.tbl_uvregression, bold_italicize_labels_levels, bold_p.tbl_stack, bold_p.tbl_uvregression, inline_text.tbl_uvregression, modify_header, tbl_merge, tbl_stack, tbl_uvregression

Examples

```
tbl_uvglm_sort_p_ex <-
  trial %>%
  dplyr::select(age, marker, response, grade) %>%
  tbl_uvregression(
    method = glm,
    y = response,
    method.args = list(family = binomial),
    exponentiate = TRUE
  ) %>%
  sort_p()
```

---

style_percent *Style percentages to be displayed in tables or text*

Description

Style percentages to be displayed in tables or text

Usage

```
style_percent(x, symbol = FALSE)
```

Arguments

x	numeric vector of percentages
symbol	Logical indicator to include percent symbol in output. Default is FALSE.

Value

A character vector of styled percentages

Author(s)

Daniel D. Sjoberg

Examples

```
percent_vals <- c(-1, 0, 0.0001, 0.005, 0.01, 0.10, 0.45356, 0.99, 1.45)
style_percent(percent_vals)
style_percent(percent_vals, symbol = TRUE)
```

`style_pvalue` *Style p-values to be displayed in tables or text*

Description

Style p-values to be displayed in tables or text

Usage

```
style_pvalue(x, digits = 1, prepend_p = FALSE)
```

Arguments

`x`	Numeric vector of p-values.
`digits`	Number of digits large p-values are rounded. Must be 1 or 2. Default is 1.
`prepend_p`	Logical. Should 'p=' be prepended to formatted p-value. Default is `FALSE`

Value

A character vector of styled p-values

Author(s)

Daniel D. Sjoberg

Examples

```
pvals <- c(
  1.5, 1, 0.999, 0.5, 0.25, 0.2, 0.197, 0.12, 0.10, 0.0999, 0.06,
  0.03, 0.002, 0.001, 0.00099, 0.0002, 0.00002, -1
)
style_pvalue(pvals)
style_pvalue(pvals, digits = 2, prepend_p = TRUE)
```

`style_ratio` *Implement significant figure-like rounding for ratios*

Description

When reporting ratios, such as relative risk or an odds ratio, we'll often want the rounding to be similar on each side of the number 1. For example, if we report an odds ratio of 0.95 with a confidence interval of 0.70 to 1.24, we would want to round to two decimal places for all values. In other words, 2 significant figures for numbers less than 1 and 3 significant figures 1 and larger. `style_ratio()` performs significant figure-like rounding in this manner.

Usage

```
style_ratio(x, digits = 2)
```

Arguments

`x`	Numeric vector
`digits`	Integer specifying the number of significant digits to display for numbers below 1. Numbers larger than 1 will be be `digits + 1`. Default is `digits = 2`.

Value

A character vector of styled ratios

Author(s)

Daniel D. Sjoberg

See Also

style_sigfig

Examples

```
c(
  0.123, 0.9, 1.1234, 12.345, 101.234, -0.123,
  -0.9, -1.1234, -12.345, -101.234
) %>%
  style_ratio()
```

`style_sigfig` *Implement significant figure-like rounding*

Description

Converts a numeric argument into a string that has been rounded to a significant figure-like number. Scientific notation output is avoided, however, and additional significant figures may be displayed for large numbers. For example, if the number of significant digits requested is 2, 123 will be displayed (rather than 120 or 1.2x10^2).

Usage

```
style_sigfig(x, digits = 2)
```

Arguments

`x`	Numeric vector
`digits`	Integer specifying the minimum number of significant digits to display

Details

If 2 sig figs are input, the number is rounded to 2 decimal places when `abs(x) < 1`, 1 decimal place when `abs(x) >= 1 & abs(x) < 10`, and to the nearest integer when `abs(x) >= 10`.

Value

A character vector of styled numbers

Author(s)

Daniel D. Sjoberg

Examples

```
c(0.123, 0.9, 1.1234, 12.345, -0.123, -0.9, -1.1234, -12.345, NA, -0.001) %>%
  style_sigfig()
```

tbl_merge *Merge two or more gtsummary regression objects*

Description

Merges two or more tbl_regression, tbl_uvregression, or tbl_stack objects and adds appropriate spanning headers.

Usage

```
tbl_merge(tbls, tab_spanner = paste0(c("Model "), seq_len(length(tbls))))
```

Arguments

tbls	List of gtsummary regression objects
tab_spanner	Character vector specifying the spanning headers. Must be same length as tblsargument

Value

A tbl_merge object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

tbl_stack

Other tbl_regression tools: add_global_p.tbl_regression, add_nevent.tbl_regression, bold_italicize_labels_le
bold_p.tbl_regression, bold_p.tbl_stack, inline_text.tbl_regression, modify_header,
sort_p.tbl_regression, tbl_regression, tbl_stack

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_nevent.tbl_uvregression,
add_q.tbl_uvregression, bold_italicize_labels_levels, bold_p.tbl_stack, bold_p.tbl_uvregression,
inline_text.tbl_uvregression, modify_header, sort_p.tbl_uvregression, tbl_stack, tbl_uvregression

Examples

```
library(survival)
t1 <-
  glm(response ~ trt + grade + age, trial, family = binomial) %>%
  tbl_regression(exponentiate = TRUE)
t2 <-
```

```
  coxph(Surv(ttdeath, death) ~ trt + grade + age, trial) %>%
  tbl_regression(exponentiate = TRUE)
tbl_merge_ex <-
  tbl_merge(
    tbls = list(t1, t2),
    tab_spanner = c("Tumor Response", "Time to Death")
  )
```

`tbl_regression` *Display regression model results in table*

Description

This function uses broom::tidy and broom.mixed::tidy to perform the initial model formatting. Review the tbl_regression vignette for detailed examples.

Usage

```
tbl_regression(x, label = NULL, exponentiate = FALSE, include = NULL,
  exclude = NULL, show_yesno = NULL, conf.level = NULL,
  intercept = FALSE, estimate_fun = NULL, pvalue_fun = NULL)
```

Arguments

`x`	Regression model object
`label`	List of formulas specifying variables labels, e.g. `list("age" ~ "Age,yrs","ptstage" ~ "Path T Stage")`
`exponentiate`	Logical indicating whether to exponentiate the coefficient estimates. Default is `FALSE`.
`include`	Names of variables to include in output.
`exclude`	Names of variables to exclude from output.
`show_yesno`	By default yes/no categorical variables are printed on a single row, when the 'No' category is the reference group. To print both levels in the output table, include the variable name in the show_yesno vector, e.g. 'show_yesno = c("var1", "var2")"
`conf.level`	Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
`intercept`	Logical argument indicating whether to include the intercept in the output. Default is `FALSE`
`estimate_fun`	Function to round and format coefficient estimates. Default is style_sigfig when the coefficients are not transformed, and style_ratio when the coefficients have been exponentiated.
`pvalue_fun`	Function to round and format p-values. Default is style_pvalue. The function must have a numeric vector input (the numeric, exact p-value), and return a string that is the rounded/formatted p-value (e.g. `pvalue_fun = function(x) style_pvalue(x,digits = 2)` or equivalently, `purrr::partial(style_pvalue,digits = 2)`).

Value

A `tbl_regression` object

Setting Defaults

If you prefer to consistently use a different function to format p-values or estimates, you can set options in the script or in the user- or project-level startup file, '.Rprofile'. The default confidence level can also be set.

- `options(gtsummary.pvalue_fun = new_function)`
- `options(gtsummary.tbl_regression.estimate_fun = new_function)`
- `options(gtsummary.conf.level = 0.90)`

Note

The N reported in the `tbl_regression()` output is the number of observations in the data frame `model.frame(x)`. Depending on the model input, this N may represent different quantities. In most cases, it is the total number of observations in your model; however, the precise definition of an observation, or unit of analysis, may differ across models. Here are some common examples.

1. Survival regression models including time dependent covariates.
2. Random- or mixed-effects regression models with clustered data.
3. GEE regression models with clustered data.

This list is not exhaustive, and care should be taken for each number reported.

Example Output

Author(s)

Daniel D. Sjoberg

See Also

See tbl_regression vignette for detailed examples

Other tbl_regression tools: `add_global_p.tbl_regression`, `add_nevent.tbl_regression`, `bold_italicize_labels_le`
`bold_p.tbl_regression`, `bold_p.tbl_stack`, `inline_text.tbl_regression`, `modify_header`,
`sort_p.tbl_regression`, `tbl_merge`, `tbl_stack`

Examples

```
library(survival)
tbl_regression_ex1 <-
  coxph(Surv(ttdeath, death) ~ age + marker, trial) %>%
  tbl_regression(exponentiate = TRUE)

tbl_regression_ex2 <-
```

```
  glm(response ~ age + grade, trial, family = binomial(link = "logit")) %>%
  tbl_regression(exponentiate = TRUE)

library(lme4)
tbl_regression_ex3 <-
  glmer(am ~ hp + (1 | gear), mtcars, family = binomial) %>%
  tbl_regression(exponentiate = TRUE)
```

`tbl_stack` *Stacks two or more gtsummary regression objects*

Description

Assists in patching together more complex tables. `tbl_stack()` appends two or more `tbl_regression` or `tbl_merge` objects. gt attributes from the first regression object are utilized for output table. If combining `tbl_stack()` and `tbl_merge()`, merge first then stack, or stack first and then merge.

Usage

```
tbl_stack(tbls)
```

Arguments

`tbls`	List of gtsummary regression objects

Value

A `tbl_stack` object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

tbl_merge

Other tbl_regression tools: add_global_p.tbl_regression, add_nevent.tbl_regression, bold_italicize_labels_le
bold_p.tbl_regression, bold_p.tbl_stack, inline_text.tbl_regression, modify_header,
sort_p.tbl_regression, tbl_merge, tbl_regression

Other tbl_uvregression tools: add_global_p.tbl_uvregression, add_nevent.tbl_uvregression,
add_q.tbl_uvregression, bold_italicize_labels_levels, bold_p.tbl_stack, bold_p.tbl_uvregression,
inline_text.tbl_uvregression, modify_header, sort_p.tbl_uvregression, tbl_merge, tbl_uvregression

Examples

```
# Example 1 - stacking two tbl_regression objects
t1 <-
  glm(response ~ trt, trial, family = binomial) %>%
  tbl_regression(
    exponentiate = TRUE,
    label = list(vars(trt) ~ "Treatment (unadjusted)")
  )

t2 <-
  glm(response ~ trt + grade + stage + marker, trial, family = binomial) %>%
  tbl_regression(
    include = "trt",
    exponentiate = TRUE,
    label = list(vars(trt) ~ "Treatment (adjusted)")
  )

tbl_stack_ex1 <- tbl_stack(list(t1, t2))

# Example 2 - stacking two tbl_merge objects
library(survival)
t3 <-
  coxph(Surv(ttdeath, death) ~ trt, trial) %>%
  tbl_regression(
    exponentiate = TRUE,
    label = list(vars(trt) ~ "Treatment (unadjusted)")
  )

t4 <-
  coxph(Surv(ttdeath, death) ~ trt + grade + stage + marker, trial) %>%
  tbl_regression(
    include = "trt",
    exponentiate = TRUE,
    label = list(vars(trt) ~ "Treatment (adjusted)")
  )


# first merging, then stacking
row1 <- tbl_merge(list(t1, t3), tab_spanner = c("Tumor Response", "Death"))
row2 <- tbl_merge(list(t2, t4))
tbl_stack_ex2 <-
  tbl_stack(list(row1, row2))
```

---

tbl_summary *Create a table of summary statistics*

Description

The `tbl_summary` function calculates descriptive statistics for continuous, categorical, and dichotomous variables. Review the tbl_summary vignette for detailed examples.

Usage

```
tbl_summary(data, by = NULL, label = NULL, statistic = NULL,
  digits = NULL, type = NULL, value = NULL, group = NULL,
  missing = c("ifany", "always", "no"), missing_text = "Unknown",
  sort = NULL, percent = c("column", "row", "cell"))
```

Arguments

data	A data frame
by	A column name in data. Summary statistics will be calculated separately for each level of the by variable (e.g. by = trt). If NULL, summary statistics are calculated using all observations.
label	List of formulas specifying variables labels, e.g. list(vars(age) ~ "Age,yrs",vars(ptstage) ~ "Path T Stage"). If a variable's label is not specified here, the function will take the label attribute (attr(data$age,"label")). If attribute label is NULL, the variable name will be used.
statistic	List of formulas specifying types of summary statistics to display for each variable. The default is list(all_continuous() ~ "{median} ({p25},{p75})",all_categorical() ~ "{n} ({p}%)"). See below for details.
digits	List of formulas specifying the number of decimal places to round continuous summary statistics. If not specified, tbl_summary guesses an appropriate number of decimals to round statistics. When multiple statistics are displayed for a single variable, supply a vector rather than an integer. For example, if the statistic being calculated is "{mean} ({sd})" and you want the mean rounded to 1 decimal place, and the SD to 2 use digits = list("age" ~ c(1,2)).
type	List of formulas specifying variable types. Accepted values are c("continuous","categorical","dich e.g. type = list(starts_with(age) ~ "continuous","female" ~ "dichotomous"). If type not specified for a variable, the function will default to an appropriate summary type. See below for details.
value	List of formulas specifying the value to display for dichotomous variables. See below for details.
group	DEPRECATED. Previously required to work with the 'group' argument in add_p. No longer required.
missing	Indicates whether to include counts of NA values in the table. Allowed values are "no" (never display NA values), "ifany" (only display if any NA values), and "always" (includes NA count row for all variables). Default is "ifany".
missing_text	String to display for count of missing observations. Default is "Unknown".
sort	List of formulas specifying the type of sorting to perform for categorical data. Options are frequency where results are sorted in descending order of frequency and alphanumeric, e.g. sort = list(everything() ~ "frequency")
percent	Indicates the type of percentage to return. Must be one of "column", "row", or "cell". Default is "column".

Value

A tbl_summary object

select helpers

Select helpers from the {tidyselect} package and {gtsummary} package are available to modify default behavior for groups of variables. For example, by default continuous variables are reported with the median and IQR. To change all continuous variables to mean and standard deviation use `statistic = list(all_continuous() ~ "{mean} ({sd})")`.

All columns with class logical are displayed as dichotomous variables showing the proportion of events that are `TRUE` on a single row. To show both rows (i.e. a row for `TRUE` and a row for `FALSE`) use `type = list(all_logical() ~ "categorical")`.

The select helpers are available for use in any argument that accepts a list of formulas (e.g. `statistic`, `type`, `digits`, `value`, `sort`, etc.)

statistic argument

The statistic argument specifies the statistics presented in the table. The input is a list of formulas that specify the statistics to report. For example, `statistic = list("age" ~ "{mean} ({sd})")` would report the mean and standard deviation for age; `statistic = list(all_continuous() ~ "{mean} ({sd})")` would report the mean and standard deviation for all continuous variables. A statistic name that appears between curly brackets will be replaced with the numeric statistic (see glue::glue).

For categorical variables the following statistics are available to display.

- `{n}` frequency
- `{N}` denominator, or cohort size
- `{p}` formatted percentage

For continuous variables the following statistics are available to display.

- `{median}` median
- `{mean}` mean
- `{sd}` standard deviation
- `{var}` variance
- `{min}` minimum
- `{max}` maximum
- `{p##}` any integer percentile, where ## is an integer from 0 to 100
- `{foo}` any function of the form `foo(x)` is accepted where `x` is a numeric vector

type argument

tbl_summary displays summary statistics for three types of data: continuous, categorical, and dichotomous. If the type is not specified, tbl_summary will do its best to guess the type. Dichotomous variables are categorical variables that are displayed on a single row in the output table, rather than one row per level of the variable. Variables coded as TRUE/FALSE, 0/1, or yes/no are assumed to be dichotomous, and the TRUE, 1, and yes rows will be displayed. Otherwise, the value to display must be specified in the `value` argument, e.g. `value = list("varname" ~ "level to show")`

Example Output

Author(s)

Daniel D. Sjoberg

See Also

See tbl_summary vignette for detailed examples

Other tbl_summary tools: add_n, add_overall, add_p, add_q.tbl_summary, add_stat_label, bold_italicize_labels_levels, bold_p.tbl_summary, inline_text.tbl_summary, modify_header, sort_p.tbl_summary

Examples

```
tbl_summary_ex1 <-
  trial %>%
  dplyr::select(age, grade, response) %>%
  tbl_summary()

tbl_summary_ex2 <-
  trial %>%
  dplyr::select(age, grade, response, trt) %>%
  tbl_summary(
    by = trt,
    label = list(vars(age) ~ "Patient Age"),
    statistic = list(all_continuous() ~ "{mean} ({sd})"),
    digits = list(vars(age) ~ c(0, 1))
  )
```

---

tbl_survival *Creates table of univariate summary statistics for time-to-event endpoints*

Description

Creates table of univariate summary statistics for time-to-event endpoints

Usage

```
tbl_survival(x, ...)
```

Arguments

x	A survfit object
...	Additional arguments passed to other methods

See Also

tbl_survival.survfit

`tbl_survival.survfit` *Creates table of survival probabilities*

Description

Function takes a `survfit` object as an argument, and provides a formatted summary of the results

Usage

```
## S3 method for class 'survfit'
tbl_survival(x, times = NULL, probs = NULL,
  label = ifelse(is.null(probs), "{time}", "{prob*100}%"),
  level_label = "{level}, N = {n}", header_label = NULL,
  header_estimate = NULL, failure = FALSE, missing = "-",
  estimate_fun = NULL, ...)
```

Arguments

`x`	A survfit object with a no stratification (e.g. `survfit(Surv(ttdeath,death) ~ 1,trial)`), or a single stratifying variable (e.g. `survfit(Surv(ttdeath,death) ~ trt,trial)`)
`times`	Numeric vector of times for which to return survival probabilities.
`probs`	Numeric vector of probabilities with values in (0,1) specifying the survival quantiles to return
`label`	String defining the label shown for the time or prob column. Default is `"{time}"` or `"{prob*100}%"`. The input uses glue::glue notation to convert the string into a label. A common label may be `"{time} Months"`, which would resolve to "6 Months" or "12 Months" depending on specified `times`.
`level_label`	Used when survival results are stratified. It is a string defining the label shown. The input uses glue::glue notation to convert the string into a label. The default is `"{level}, N = {n}"`. Other information available to call are `'{n}'`, `'{level}'`, `'{n.event.tot}'`, `'{n.event.strata}'`, and `'{strata}'`. See below for details.
`header_label`	String to be displayed as column header. Default is `'**Time**'` when `time` is specified, and `'**Quantile**'` when `probs` is specified.
`header_estimate`	String to be displayed as column header of the Kaplan-Meier estimate. Default is `'**Probability**'` when `time` is specified, and `'**Time**'` when `probs` is specified.
`failure`	Calculate failure probabilities rather than survival probabilities. Default is `FALSE`. Does NOT apply to survival quantile requests

missing	String indicating what to replace missing confidence limits with in output. Default is missing = "-"
estimate_fun	Function used to format the estimate and confidence limits. The default is style_percent(x,symbol = TRUE) for survival probabilities, and style_sigfig(x,digits = 3) for time estimates.
...	Not used

Value

A tbl_survival object

level_label argument

The level_label is used to modify the stratum labels. The default is level_label = "{level},N = {n}". The quantities in the curly brackets evaluate to stratum-specific values. For example, in the trial data set, there is a column called trt with levels 'Drug' and 'Placebo'. In this example, {level} would evaluate to either 'Drug' or 'Placebo' depending on the stratum. Other quantities available to print are:

- {level} level of the stratification variable
- {level_label} label of level for the stratification variable
- {n} number of observations, or number within stratum
- {n.event.tot} total number of events (total across stratum, if applicable)
- {n.event.strata} total number of events within stratum, if applicable
- {strata} raw stratum specification from survfit object

Example Output

Author(s)

Daniel D. Sjoberg

See Also

Other tbl_survival tools: inline_text.tbl_survival, modify_header

Examples

```
library(survival)
fit1 <- survfit(Surv(ttdeath, death) ~ trt, trial)
tbl_strata_ex1 <-
  tbl_survival(fit1,
    times = c(12, 24),
    label = "{time} Months"
  )

fit2 <- survfit(Surv(ttdeath, death) ~ 1, trial)
```

```
tbl_nostrata_ex2 <-
  tbl_survival(fit2,
    probs = c(0.1, 0.2),
    header_estimate = "**Months**"
  )
```

`tbl_uvregression` *Display univariate regression model results in table*

Description

The `tbl_uvregression` function arguments are similar to the tbl_regression arguments. Review the tbl_uvregression vignette for detailed examples.

Usage

```
tbl_uvregression(data, method, y, method.args = NULL,
  formula = "{y} ~ {x}", exponentiate = FALSE, label = NULL,
  hide_n = FALSE, show_yesno = NULL, conf.level = NULL,
  estimate_fun = NULL, pvalue_fun = NULL)
```

Arguments

`data`	Data frame to be used in univariate regression modeling. Data frame includes the outcome variable(s) and the independent variables.
`method`	Regression method (e.g. lm, glm, survival::coxph, and more).
`y`	Model outcome (e.g. `y = recurrence` or `y = Surv(time,recur)`)
`method.args`	List of additional arguments passed on to the regression function defined by `method`.
`formula`	String of the model formula. Uses glue::glue syntax. Default is `"{y} ~ {x}"`, where `{y}` is the dependent variable, and `{x}` represents a single covariate. For a random intercept model, the formula may be `formula = "{y} ~ {x} + (1 | gear)"`.
`exponentiate`	Logical indicating whether to exponentiate the coefficient estimates. Default is `FALSE`.
`label`	List of formulas specifying variables labels, e.g. `list("age" ~ "Age,yrs","ptstage" ~ "Path T Stage")`
`hide_n`	Hide N column. Default is `FALSE`
`show_yesno`	By default yes/no categorical variables are printed on a single row, when the 'No' category is the reference group. To print both levels in the output table, include the variable name in the show_yesno vector, e.g. 'show_yesno = c("var1", "var2")"
`conf.level`	Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.

estimate_fun	Function to round and format coefficient estimates. Default is style_sigfig when the coefficients are not transformed, and style_ratio when the coefficients have been exponentiated.
pvalue_fun	Function to round and format p-values. Default is style_pvalue. The function must have a numeric vector input (the numeric, exact p-value), and return a string that is the rounded/formatted p-value (e.g. `pvalue_fun = function(x) style_pvalue(x, digits = 2)` or equivalently, `purrr::partial(style_pvalue, digits = 2)`).

Value

A `tbl_uvregression` object

Setting Defaults

If you like to consistently use a different function to format p-values or estimates, you can set options in the script or in the user- or project-level startup file, '.Rprofile'. The default confidence level can also be set. Please note the default option for the estimate is the same as it is for `tbl_regression()`.

- `options(gtsummary.pvalue_fun = new_function)`
- `options(gtsummary.tbl_regression.estimate_fun = new_function)`
- `options(gtsummary.conf.level = 0.90)`

Note

The N reported in the `tbl_uvregression()` output is the number of observations in the data frame `model.frame(x)`. Depending on the model input, this N may represent different quantities. In most cases, it is the number of people or units in your model. Here are some common exceptions.

1. Survival regression models including time dependent covariates.
2. Random- or mixed-effects regression models with clustered data.
3. GEE regression models with clustered data.

This list is not exhaustive, and care should be taken for each number reported.

Example Output

Author(s)

Daniel D. Sjoberg

See Also

See tbl_regression vignette for detailed examples

Other tbl_uvregression tools: `add_global_p.tbl_uvregression`, `add_nevent.tbl_uvregression`, `add_q.tbl_uvregression`, `bold_italicize_labels_levels`, `bold_p.tbl_stack`, `bold_p.tbl_uvregression`, `inline_text.tbl_uvregression`, `modify_header`, `sort_p.tbl_uvregression`, `tbl_merge`, `tbl_stack`

Examples

```
tbl_uv_ex1 <-
  tbl_uvregression(
    trial %>% dplyr::select(response, age, grade),
    method = glm,
    y = response,
    method.args = list(family = binomial),
    exponentiate = TRUE
  )

# rounding pvalues to 2 decimal places
library(survival)
tbl_uv_ex2 <-
  tbl_uvregression(
    trial %>% dplyr::select(ttdeath, death, age, grade, response),
    method = coxph,
    y = Surv(ttdeath, death),
    label = list(vars(grade) ~ "Grade"),
    exponentiate = TRUE,
    pvalue_fun = function(x) style_pvalue(x, digits = 2)
  )
```

---

trial *Results from a simulated trial of Placebo vs Drug*

Description

A dataset containing the baseline characteristics of 200 patients randomized to Placebo or Drug. Dataset also contains the trial outcome: tumor response to the treatment.

Usage

```
trial
```

Format

A data frame with 200 rows–one row per patient

trt Treatment Randomization

age Age, yrs

marker Marker Level, ng/mL

stage T Stage

grade Grade

response Tumor Response

death Patient Died

ttdeath Months to Death/Censor

Index