

Package: utile.tables (via r-universe)

October 10, 2024

Title Build Tables for Publication

Version 0.3.0

Description Functions for building customized ready-to-export tables for publication.

License LGPL (>= 2)

URL <https://efinite.github.io/utile.tables/>

BugReports <https://github.com/efinite/utile.tables/issues>

Encoding UTF-8

Depends R (>= 3.4.0)

Imports purrr (>= 1.0.0), rlang, tidyselect, tibble, vctrs, utile.tools (>= 0.3.0)

Suggests dplyr, survival

RoxygenNote 7.2.3

Repository <https://efinite.r-universe.dev>

RemoteUrl <https://github.com/efinite/utile.tables>

RemoteRef HEAD

RemoteSha 914f66ab2574fcd651d8ed93d3831a5be7c4b8f6

Contents

build_model	2
build_model.coxph	2
build_row	4
build_table	6
build_table.coxph	7
build_table.data.frame	8
build_table.lm	10

Index	12
--------------	-----------

`build_model`*Build models*

Description

Models specified terms in model data against an existing model and returns a clean, human readable table of summarizing the effects and statistics for the newly generated model. This function is meant to simplify fitting a large number of variables against a set of time-to-event data.

Usage

```
build_model(.object, ...)
```

Arguments

`.object` An object of a supported class. See S3 methods below.
`...` Arguments passed to the appropriate S3 method.

Value

An object of class `tbl_df` (tibble) summarizing the provided object.

See Also

[build_model.coxph](#)

`build_model.coxph`*Build Cox PH models*

Description

Models specified terms in model data against an existing model and returns a clean, human readable table of summarizing the effects and statistics for the newly generated model. This functions greatly simplifies fitting a large number of variables against a set of time-to-event data.

Usage

```
## S3 method for class 'coxph'  
build_model(  
  .object,  
  ...,  
  .mv = FALSE,  
  .test = c("LRT", "Wald"),  
  .col.test = FALSE,  
  .level = 0.95,  
  .stat.pct.sign = TRUE,
```

```
.digits = 1,
.p.digits = 4
)
```

Arguments

.object	An object of class <code>coxph</code> .
...	One or more unquoted expressions separated by commas representing columns in the model data.frame. May be specified using tidyselect helpers .
.mv	A logical. Fit all terms into a single multivariable model. If left FALSE, all terms are fit in their own univariate models.
.test	A character. The name of a <code>stats::drop1</code> test to use with the model.
.col.test	A logical. Append a columns for the test and accompanying statistic used to derive the p-value.
.level	A double. The confidence level required.
.stat.pct.sign	A logical. Paste a percent symbol after all reported frequencies.
.digits	An integer. The number of digits to round numbers to.
.p.digits	An integer. The number of p-value digits to report. Note that the p-value still rounded to the number of digits specified in <code>.digits</code> .

Value

An object of class `data.frame` summarizing the provided object. If the `tibble` package has been installed, a `tibble` will be returned.

See Also

[build_model](#)

Examples

```
library(survival)
library(dplyr)

data_lung <- lung |>
  mutate_at(vars(inst, status, sex), as.factor) |>
  mutate(status = case_when(status == 1 ~ 0, status == 2 ~ 1))

fit <- coxph(Surv(time, status) ~ 1, data = data_lung)

# Create a univariate model for each variable
fit |> build_model(sex, age)
```

build_row	<i>Build summary rows</i>
-----------	---------------------------

Description

Summarize data into a data.frame with row(s). Includes optional stratification and null hypothesis testing using a factor or logical variable.

Usage

```
build_row(x, ...)  
  
## S3 method for class 'data.frame'  
build_row(  
  x,  
  y = NA_real_,  
  label = NULL,  
  label.stat = TRUE,  
  stat.pct.sign = FALSE,  
  col.overall = TRUE,  
  col.missing = FALSE,  
  col.test = FALSE,  
  digits = 1,  
  ...  
)  
  
## S3 method for class 'numeric'  
build_row(  
  x,  
  y = NA_real_,  
  label = NULL,  
  label.stat = TRUE,  
  stat = c("mean", "median"),  
  stat.pct.sign = FALSE,  
  col.overall = TRUE,  
  col.missing = FALSE,  
  test = c("anova", "kruskal", "wilcoxon"),  
  col.test = FALSE,  
  digits = 1,  
  p.digits = 4,  
  ...  
)  
  
## S3 method for class 'logical'  
build_row(  
  x,  
  y = NA_real_,
```

```

    label = NULL,
    label.stat = TRUE,
    inverse = FALSE,
    stat.pct.sign = FALSE,
    col.overall = TRUE,
    col.missing = FALSE,
    test = c("chisq", "fisher"),
    test.simulate.p = FALSE,
    col.test = FALSE,
    digits = 1,
    p.digits = 4,
    ...
)

## S3 method for class 'factor'
build_row(
  x,
  y = NA_real_,
  label = NULL,
  label.stat = TRUE,
  stat.pct.sign = FALSE,
  col.overall = TRUE,
  col.missing = FALSE,
  test = c("chisq", "fisher"),
  test.simulate.p = FALSE,
  col.test = FALSE,
  digits = 1,
  p.digits = 4,
  ...
)

```

Arguments

<code>x</code>	A data.frame, numeric, factor, or logical. Data to summarize.
<code>...</code>	Arguments passed to the appropriate S3 method.
<code>y</code>	A factor or logical. Data to optionally stratify <code>x</code> by.
<code>label</code>	A character. A label for the summarized data.
<code>label.stat</code>	A logical. Append the summary statistic used to the label.
<code>stat.pct.sign</code>	A logical. Paste a percentage symbol with each frequency. frequency.
<code>col.overall</code>	A logical. Append a column with the statistic for all data. If <code>y</code> is not specified, this parameter is ignored.
<code>col.missing</code>	A logical. Append a column with counts of missing data.
<code>col.test</code>	A logical. Append a column with the name of the statistical test used.
<code>digits</code>	An integer. Number of digits to round to.
<code>stat</code>	A character. Name of the summary statistic to use. Supported options include the mean ('mean') and median ('median') for continuous data.

test	A character. Name of statistical test to compare groups. Supported options: [continuous data] ANOVA linear model ('anova'), Kruskal-Wallis ('kruskal'), and Wilcoxon rank sum ('wilcoxon') tests; [nominal data] Pearson's Chi-squared Test ('chisq') and Fisher's Exact Test ('fisher').
p.digits	An integer. Number of p-value digits to report.
inverse	A logical. For logical data, report frequencies of the FALSE values instead.
test.simulate.p	A logical. Whether to use Monte Carlo simulation of the p-value when testing nominal data.

Value

An object of class `tbl_df` (tibble) summarizing the provided data.

Examples

```
strata <- as.factor(datasets::mtcars$cyl)

# Create a "count" row from a data.frame for a factor
build_row(x = datasets::mtcars, y = strata)

# Create a row summarizing a numeric by a factor
build_row(label = 'MPG', x = as.numeric(datasets::mtcars$mpg), y = strata)

# Create a row summarizing a logical by a factor
build_row(label = 'VS', x = as.logical(datasets::mtcars$vs), y = strata)

# Create a row summarizing a factor by a factor
build_row(label = 'Carb', x = as.factor(datasets::mtcars$carb), y = strata)
```

build_table

Build summary tables

Description

Takes a data or model object and summarizes it into a ready to export, human-readable summary table.

Usage

```
build_table(.object, ...)
```

Arguments

.object	An object of a supported class. See S3 methods below.
...	Arguments passed to the appropriate S3 method.

Value

An object of class `tbl_df` (tibble) summarizing the provided object.

See Also

[build_table.data.frame](#), [build_table.coxph](#), [build_table.lm](#)

<code>build_table.coxph</code>	<i>Build summary tables from coxph model objects</i>
--------------------------------	--

Description

Takes a Cox PH model object and summarizes it into a ready to export, human-readable summary table.

Usage

```
## S3 method for class 'coxph'
build_table(
  .object,
  ...,
  .test = c("LRT", "Wald"),
  .col.test = FALSE,
  .level = 0.95,
  .stat.pct.sign = TRUE,
  .digits = 1,
  .p.digits = 4
)
```

Arguments

<code>.object</code>	An object of class coxph .
<code>...</code>	One or more unquoted expressions separated by commas representing columns in the <code>data.frame</code> . May be specified using tidyselect helpers . If left empty, all terms are summarized.
<code>.test</code>	A character. The name of the <code>stats::drop1</code> test to use with the model. Supported tests include Wald's Test ('Wald') and Likelihood Ratio Test ('LRT').
<code>.col.test</code>	A logical. Append a columns for the test and accompanying statistic used to derive the p-value.
<code>.level</code>	A double. The confidence level required.
<code>.stat.pct.sign</code>	A logical. Paste a percent symbol after all reported frequencies.
<code>.digits</code>	An integer. The number of digits to round numbers to.
<code>.p.digits</code>	An integer. The number of p-value digits to report. Note that the p-value still rounded to the number of digits specified in <code>.digits</code> .

Value

An object of class `tbl_df` (tibble) summarizing the provided object.

See Also

[build_table](#)

Examples

```
library(survival)
library(dplyr)

data_lung <- lung |>
  mutate_at(vars(inst, status, sex), as.factor) |>
  mutate(status = case_when(status == 1 ~ 0, status == 2 ~ 1))

fit <- coxph(Surv(time, status) ~ sex + meal.cal, data = data_lung)

fit |> build_table(Sex = sex, Calories = meal.cal, .test = 'LRT')
```

`build_table.data.frame`

Build summary tables from data.frame objects

Description

Takes a `data.frame` object and summarizes the columns into a ready to export, human-readable summary table. Capable of stratifying data and performing appropriate hypothesis testing.

Usage

```
## S3 method for class 'data.frame'
build_table(
  .object,
  ...,
  .by,
  .inverse = FALSE,
  .label.stat = TRUE,
  .stat = c("mean", "median"),
  .stat.pct.sign = FALSE,
  .col.overall = TRUE,
  .col.missing = FALSE,
  .test.continuous = c("anova", "kruskal", "wilcoxon"),
  .test.nominal = c("chisq", "fisher"),
  .test.simulate.p = FALSE,
  .col.test = FALSE,
  .digits = 1,
  .p.digits = 4
)
```


Arguments

<code>.object</code>	A data.frame.
<code>...</code>	One or more unquoted expressions separated by commas representing columns in the data.frame. May be specified using tidyselect helpers . If left empty, all columns are summarized.
<code>.by</code>	An unquoted expression. The data column to stratify the summary by.
<code>.inverse</code>	A logical. For logical data, report the frequency of FALSE values instead of the TRUE.
<code>.label.stat</code>	A logical. Append the type of summary statistic to the column label.
<code>.stat</code>	A character. Name of the summary statistic to use for numeric data. Supported options include the mean ('mean') and median ('median').
<code>.stat.pct.sign</code>	A logical. Paste a percent symbol after all reported frequencies.
<code>.col.overall</code>	A logical. Append a column with the statistic for all data. If <code>.by</code> is not specified, this parameter is ignored.
<code>.col.missing</code>	A logical. Append a column listing the frequencies of missing data for each row.
<code>.test.continuous</code>	A character. Name of statistical test to compare groups. Supported options include ANOVA linear model ('anova'), Kruskal-Wallis ('kruskal'), and Wilcoxon rank sum ('wilcoxon') tests.
<code>.test.nominal</code>	A character. Name of statistical test to compare groups. Supported options include Pearson's Chi-squared Test ('chisq') and Fisher's Exact Test ('fisher').
<code>.test.simulate.p</code>	A logical. Whether to use Monte Carlo simulation of the p-value when testing nominal data.
<code>.col.test</code>	A logical. Append a column containing the test each p-value was derived from.
<code>.digits</code>	An integer. The number of digits to round numbers to.
<code>.p.digits</code>	An integer. The number of p-value digits to report.

Value

An object of class `tbl_df` (tibble) summarizing the provided object.

See Also

[build_table](#)

Examples

```
# Sample data
df <- data.frame(
  strata = factor(sample(letters[2:3], 1000, replace = TRUE)),
  numeric = sample(1:100, 1000, replace = TRUE),
  numeric2 = sample(1:100, 1000, replace = TRUE),
  factor = factor(sample(1:5, 1000, replace = TRUE)),
  logical = sample(c(TRUE,FALSE), 1000, replace = TRUE)
```

```

)

# Summarize all columns
build_table(df, .by = strata)

# Summarize & rename selected columns
build_table(df, numeric2, factor, .by = strata)

```

 build_table.lm

Build summary tables from lm model objects

Description

Takes a linear regression model object and summarizes it into a ready to export, human-readable summary table.

Usage

```

## S3 method for class 'lm'
build_table(
  .object,
  ...,
  .test = c("F", "Chisq"),
  .col.test = FALSE,
  .level = 0.95,
  .stat.pct.sign = TRUE,
  .digits = 1,
  .p.digits = 4
)

```

Arguments

<code>.object</code>	An object of class <code>lm</code> .
<code>...</code>	One or more unquoted expressions separated by commas representing columns in the data.frame. May be specified using tidyselect helpers . If left empty, all terms are summarized.
<code>.test</code>	A character. The name of the <code>stats::drop1</code> test to use with the model. Supported options include the F-Test ('F') and Chi-squared Test ('Chisq').
<code>.col.test</code>	A logical. Append a columns for the test and accompanying statistic used to derive the p-value.
<code>.level</code>	A double. The confidence level required.
<code>.stat.pct.sign</code>	A logical. Paste a percent symbol after all reported frequencies.
<code>.digits</code>	An integer. The number of digits to round numbers to.
<code>.p.digits</code>	An integer. The number of p-value digits to report. Note that the p-value still rounded to the number of digits specified in <code>.digits</code> .

Value

An object of class `tbl_df` (tibble) summarizing the provided object.

See Also

[build_table](#)

Examples

```
library(dplyr)

data_mtcars <- datasets::mtcars |>
  mutate_at(vars('vs', 'am'), as.logical) |>
  mutate_at(vars('gear', 'carb', 'cyl'), as.factor)

fit <- lm(mpg ~ vs + drat + cyl, data = data_mtcars)

fit |> build_table()
```

Index

`build_model`, [2](#), [3](#)
`build_model.coxph`, [2](#), [2](#)
`build_row`, [4](#)
`build_table`, [6](#), [8](#), [9](#), [11](#)
`build_table.coxph`, [7](#), [7](#)
`build_table.data.frame`, [7](#), [8](#)
`build_table.lm`, [7](#), [10](#)

`coxph`, [3](#), [7](#)

`lm`, [10](#)

`stats::drop1`, [3](#), [7](#), [10](#)