Package ‘utile.tables’

December 2, 2019

Title Build Tables for Publication
Version 0.1.8
Description A collection of functions to make building customized ready-to-export tables for publication purposes easier and creating summaries of large datasets for review a breeze. Includes methods for automatically building a table from data or building the table row-by-row.
License LGPL (>= 2)
URL https://github.com/efinite/utile.tables
BugReports https://github.com/efinite/utile.tables/issues
Encoding UTF-8
LazyData TRUE
Depends R (>= 3.4.0)
Imports utile.tools (>= 0.2.3), tibble, dplyr, rlang, survival, purrr
RoxygenNote 7.0.1
NeedsCompilation no
Author Eric Finnesgard [aut, cre], Jennifer Grauberger [aut]
Maintainer Eric Finnesgard <finnesgard.eric@mayo.edu>
Repository CRAN
Date/Publication 2019-12-02 10:20:02 UTC

R topics documented:

  build_event_row ........................................... 2
  build_event_row_ ........................................... 3
  build_event_table ........................................... 4
  build_footer ............................................... 5
  build_row ................................................... 6
  build_row_ ................................................... 8
  build_table ............................................... 10

Index 12
build_event_row

Description

Creates a tibble row summarizing a predictor (column) in a given time-to-event model.

Usage

```r
build_event_row(
  .table = NULL,
  label = NULL,
  col = NULL,
  fit = NULL,
  percent.sign = TRUE,
  digits = 1,
  p.digits = 4,
  indent = FALSE
)
```

Arguments

- `.table`: Optional. Tibble. A tibble for row to be appended.
- `label`: Optional. Character. Row name to print in the table. Defaults to value of `col` parameter.
- `col`: Required. Character. Name of column used as a parameter in the time-to-event function.
- `fit`: Required. `survival::coxph()`.
- `percent.sign`: Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
- `digits`: Optional. Integer. Number of digits to round numerics to. Defaults to 1.
- `p.digits`: Optional. Integer. Number of digits to print for p-value. Note that p-values are still rounded based on the `digits` parameter. Defaults to 4.
- `indent`: Optional. Logical. Indent a variable labels. Defaults to FALSE.

Value

Data is returned in the form of a tibble containing a row for the specified parameter.

Examples

```r
library(survival)
library(dplyr)

data_lung <- lung %>%
  as_tibble() %>%
```

mutate_at(vars(inst, status, sex), as.factor) %>%
mutate(status = case_when(status == 1 ~ 0, status == 2 ~ 1))

# Stand-alone row
build_event_row(
  label = 'Meal calories',
  col = 'meal.cal',
  fit = coxph(Surv(time, status) ~ meal.cal, data = data_lung)
)

# Build a table row-by-row
build_event_row(
  label = 'Age, years',
  col = 'age',
  fit = coxph(Surv(time, status) ~ age, data = data_lung)
) %>%
build_event_row(
  label = 'Sex',
  col = 'sex',
  fit = coxph(Surv(time, status) ~ sex, data = data_lung)
) %>%
build_event_row(
  label = 'Institution',
  col = 'inst',
  fit = coxph(Surv(time, status) ~ inst, data = data_lung)
)

---

**build_event_row_**  
**build_event_row_**

**Description**

A factory for creating a copy of `build_event_row()` with built in data, `fit`, and customized defaults.

**Usage**

`build_event_row_(fit, data, percent.sign, digits, p.digits, indent)`

**Arguments**

- **fit**: Required. Formula, `survival::survfit()`, `survival::coxph()`. The formula must contain a `survival::Surv()` object as first term. All terms must be present in `data`.
- **data**: Semi-optional. Tibble. Contains data for time-to-event model. Only required if `fit` is a formula.
- **percent.sign**: Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
- **digits**: Optional. Integer. Number of digits to round numerics to. Defaults to 1.
- **p.digits**: Optional. Integer. Number of digits to print for `p`-values. Note that `p`-values are still rounded based on `digits` parameter. Defaults to 4.
- **indent**: Optional. Logical. Indent a variable labels. Defaults to FALSE.
Value
A custom build_event_row() function. See related documentation for behavior.

Examples
library(survival)
library(dplyr)

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

row <- build_event_row_(
  Surv(time, status) ~ 1,
  data = data_lung,
  digits = 2
)

row(label = "Age, years", col = "age") %>%
row(label = "Sex", col = "sex") %>%
row(label = "Institution", col = "inst")

Description
Creates time-to-event models in an automated fashion and summarizes them in a tibble.

Usage
build_event_table(fit, data, cols, skip, mv, percent.sign, digits, p.digits)

Arguments

fit Required. Formula, survival::survfit(), survival::coxph(). The formula must contain a survival::Surv() object as first term. All terms must be present in data.
data Semi-optional. Tibble. Contains data for time-to-event model. Only required if \fit is a formula.
cols Optional. Character. Columns to use as predictors in time-to-event model. Defaults to all usable columns in \data.
skip Optional. Character. Names of columns to skip as part of predictor testing.
mv Optional. Logical. Indicates provided \cols should be tested as part of one multi-variate model. Defaults to FALSE (univariate; seperate models).
percent.sign Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
digits  Optional. Integer. Number of digits to round numerics to. Defaults to 1.
p.digits  Optional. Integer. Number of digits to print for p-values. Note that p-values are still rounded based on \texttt{digits} parameter. Defaults to 4.

Value

Data is returned in the form of a tibble containing a row for each parameter.

Examples

library(survival)
library(dplyr)

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

# Automatically model each parameter
build_event_table(Surv(time, status) ~ 1, skip = '\texttt{inst}', data = data_lung)

# Automatically model all parameters together
build_event_table(Surv(time, status) ~ 1, skip = '\texttt{inst}', mv = TRUE, data = data_lung)
by Optional. Character. Name of factor or logical column table is stratified by.

data Required. Tibble. Contains data being summarized.

parametric Optional. Logical. Indicates parametric testing should be used for comparisons (Fisher’s exact and Student’s Unpaired T-Test). Defaults to FALSE (non-parametric; Chi-squared and Wilcox Rank-sum).

Value
Data is returned in the form of a tibble containing the row(s).

Examples

```r
library(dplyr)

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

# Create footer row
build_footer(
  by = "am",
  data = data_mtcars,
  parametric = FALSE
)
```

Description
Generates a summarizing table row from column data.

Usage

```r
build_row(
  .table = NULL,
  label = NULL,
  col = NULL,
  by = NULL,
  data = NULL,
  parametric = FALSE,
  inverse = FALSE,
  indent = FALSE,
  percent.sign = TRUE,
  less.than.one = FALSE,
  remove.na = TRUE,
  label.stats = TRUE,
)```
**build_row**

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

**Arguments**

- **.table**: Optional. Tibble. A tibble to append row data to.
- **label**: Optional. Character. Label for the row (i.e. 'Age, years'). Defaults to value of 'col'.
- **col**: Optional. Character. Name of column to be summarized. If left blank, either a frequency row will be created (if 'label' not specified) or empty label row created (if 'label' specified).
- **by**: Optional. Character. Name of factor or logical column to stratify by. If using `build_row_()`, this may be pre-specified.
- **data**: Required/Optional. Tibble or Character. Contains data to summarize. If using `build_row_()` to pre-load data, you may instead provide a character string of code that would represent how you would reference the tibble (i.e. `.index` or `.index` you also specified a default tibble to be used via `data.default`, that tibble will automatically be used as a fallback if the 'data' parameter is left blank.
- **parametric**: Optional. Logical. Indicates parametric testing should be used for comparisons (Fisher's exact and Student's Unpaired T-Test). Defaults to FALSE (non-parametric; Chi-squared and Wilcox Rank-sum).
- **inverse**: Optional. Logical. Indicates to summarize the FALSE/No data of a logical column (i.e. 'Smoking Hx, yes' -> 'Smoking Hx, no'). Defaults to FALSE (Summarizes TRUE/Yes data).
- **indent**: Optional. Logical. Indent a variable’s label. Defaults to FALSE.
- **percent.sign**: Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
- **less.than.one**: Optional. Logical. Indicates means/medians that round to 0 should be printed as <1 (i.e. <1 [0-4]). Defaults to FALSE (0).
- **remove.na**: Optional. Logical. Remove NA from denominator in frequency calculations. Defaults to TRUE.
- **label.stats**: Optional. Logical. Whether to append the type of statistic (median, n, mean) to the row’s label. Defaults to TRUE.
- **digits**: Optional. Integer. Number of digits to round numerics to. Defaults to 1.
- **p.digits**: Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded based on 'digits' parameter. Defaults to 4.
- **...**: Optional. Any other variables or tibbles you want to make available for use as row data. Recommend naming each of these with a starting '.' to ensure they do not conflict with other variables (i.e. `build_row(.index = data.index, .radiology = data.radiology)`).

**Value**

Data is returned in the form of a tibble containing the row(s).
Examples

```r
library(dplyr)

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

# Stand-alone row
build_row(
  label = 'Gears',
  col = 'gear',
  by = 'am',
  data = data_mtcars,
  percent.sign = FALSE
)

# Summary Table
build_row(label = 'Miles per gallon', col = 'mpg', data = data_mtcars) %>%
build_row(label = 'Cylinders', col = 'cyl', data = data_mtcars) %>%
build_row(label = 'Horsepower', col = 'hp', data = data_mtcars)
```

Description

A factory function to create a copy of `build_row()` with built in data and pre-specified rules for row formatting.

Usage

```r
build_row(
  by = NULL,
  data.default = NULL,
  digits = 1,
  percent.sign = FALSE,
  inverse = FALSE,
  indent = FALSE,
  less.than.one = FALSE,
  label.stats = TRUE,
  parametric = FALSE,
  p.digits = 4,
  remove.na = TRUE,
  ...
)
```
Arguments

by:
Optional. Character or Quosure. Name of factor or logical column to stratify by or quosure code to be used by dplyr::mutate to create a grouping variable on the fly (i.e. quo(pt_id tibbles. Either must be applicable to all provided tibbles!

data.default:
Optional. Character. Name of a provided tibble to use as default data source if the \data\ parameter is not specified in build_row().

digits:
Optional. Integer. Number of digits to round numerics to. Defaults to 1.

percent.sign:
Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.

inverse:
Optional. Logical. Indicates to summarize the FALSE/No data of a logical column (i.e. 'Smoking Hx, yes' -> 'Smoking Hx, no'). Defaults to FALSE (Summarizes TRUE/Yes data).

indent:
Optional. Logical. Indent a variable labels. Defaults to FALSE.

less.than.one:
Optional. Logical. Indicates means/medians that round to 0 should be printed as <1 (i.e. <1 [0-4]). Defaults to FALSE (0).

label.stats:
Optional. Logical. Whether to append the type of statistic (median, n, mean) to the row’s label. Defaults to TRUE.

parametric:
Optional. Logical. Indicates parametric testing should be used for comparisons (Fisher’s exact and Student’s Unpaired T-Test). Defaults to FALSE (non-parametric; Chi-squared and Wilcox Rank-sum).

p.digits:
Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded based on ‘digits’ parameter. Defaults to 4.

remove.na:
Optional. Logical. Remove NA from denominator in frequency calculations. Defaults to TRUE.

...:
Optional. Any other variables or tibbles you want to make available for use in build_row(). I recommend naming each of these with a starting ‘.’ to ensure they do not conflict with other variables (i.e. build_row_(.index = data.index, .radiology = data.radiology)). Note that this is optional as a tibble may be provided to a resultant row function on the fly instead.

Value

A custom build_row() function. See documentation for behavior.

Examples

library(dplyr)

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

# Create instance of build_row() with custom defaults
row <- build_row_(
  by = 'am',
percent.sign = FALSE,
less.than.one = TRUE,
label.stat = FALSE,
num2text = TRUE,
base.table = FALSE,
# Data (more than one allowed!)
default.data = '.mtcars',
    .mtcars = data_mtcars,
    .mtcars_alt = data_mtcars
)

row() %>% # Count row
    row(label = 'Car Features') %>% # Row without data
    row(label = 'Miles per gallon', col = 'mpg') %>%
    row(col = 'cyl', data = '.mtcars %>% filter(mpg > 20)') %>% # subset of data
    row(label = 'Horsepower', col = 'hp') %>%
    row(label = 'Engine Shape', col = 'vs', percent.sign = TRUE, data = '.mtcars_alt')

---

**build_table**

**Description**

A function for summarizing columns of data. Can work in an automated fashion or with manually
specified options. It is essentially a wrapper for build_row().

**Usage**

```r
build_table(
  data = NULL,
  by = NULL,
  cols = NULL,
  skip = NULL,
  digits = 1,
  percent.sign = FALSE,
  less.than.one = FALSE,
  inverse = FALSE,
  indent = FALSE,
  parametric = FALSE,
  footer.stats = FALSE,
  p.digits = 4,
  remove.na = TRUE
)
```

**Arguments**

- **data**: Required. Tibble. Contains data to be summarized.
- **by**: Optional. Character. Name of factor or logical column to stratify summaries by.
cols  Optional. Character. Contains character names of columns to summarize. Defaults to all columns.
skip  Optional. Character. Names of columns to skip as part of predictor testing.
digits Optional. Integer. Number of digits to round numerics to. Defaults to 1.
percent.sign  Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
less.than.one  Optional. Logical indicating whether means/medians that round to 0 should be printed as <1 (i.e. <1 [0-4]). Defaults to printing the 0.
inverse  Optional. Logical. Indicates to summarize the FALSE/No data of logical columns (i.e. 'Smoking Hx, yes' -> 'Smoking Hx, no'). Defaults to FALSE (Summarizes TRUE/Yes data).
indent  Optional. Logical. Indent variable labels. Defaults to FALSE.
parametric  Optional. Logical. Indicates parametric testing should be used for comparisons (Fisher's exact and Student’s Unpaired T-Test). Defaults to FALSE (non-parametric; Chi-squared and Wilcox Rank-sum).
footer.stats  Optional. Logical. Most stats summary into a footer row. Removes the stat type from row labels. Defaults to FALSE.
p.digits Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded based on `digits` parameter. Defaults to 4.
remove.na  Optional. Logical. Remove NA from denominator in frequency calculations. Defaults to TRUE.

Value

Data is returned in the form of a tibble containing the row(s).

Examples

library(dplyr)

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

build_table(
data = data_mtcars,
by = 'vs',
cols = c('gear', 'mpg', 'carb', 'am', 'hp'),
percent.sign = FALSE,
less.than.one = TRUE,
footer.stats = TRUE)
)
Index

build_event_row, 2
build_event_row_, 3
build_event_table, 4
build_footer, 5
build_row, 6
build_row_, 8
build_table, 10