

Package ‘utile.tools’

July 1, 2019

Version 0.2.0

Title Summarize Data for Publication

Description A variety of tools for preparing and summarizing data for publication purposes. Function verbs include 'tabulate' for creating usable tabulated data from models, 'paste' for generating human-readable statistics from a variety of summarizable data types, 'calc' for reliably calculating differences between data points, and 'test' for conducting simple statistical tests which return human-readable results.

License LGPL (>= 2)

Encoding UTF-8

LazyData TRUE

Depends R (>= 3.4.0)

Imports tibble, dplyr, lubridate, rlang, purrr, stringr

Suggests survival, MASS

RoxygenNote 6.1.1

NeedsCompilation no

Author Eric Finnesgard [aut, cre]

Maintainer Eric Finnesgard <finnesgard.eric@mayo.edu>

Repository CRAN

Date/Publication 2019-07-01 08:20:03 UTC

R topics documented:

calc_duration	2
paste_efs	3
paste_freq	3
paste_mean	4
paste_median	4
tabulate_at_risk	5
tabulate_model	6
tabulate_model.coxph	6
tabulate_model.glm	7

tabulate_model.lm	8
test_factor	9
test_numeric	10

Index	11
--------------	-----------

calc_duration	<i>Calculate Duration</i>
---------------	---------------------------

Description

Returns the duration of time between two provided date objects. Supports vectorized data (i.e. `dplyr::mutate()`).

Usage

```
calc_duration(start = NA, end = NA, units = "years")
```

Arguments

start	Required. Date or POSIXt object. The start date/timestamp.
end	Required. Date or POSIXt object. The end date/timestamp.
units	Optional. Character. Units of the returned duration (i.e. 'seconds', 'days', 'years'). Defaults to 'years'.

Examples

```
# Timestamps
calc_duration(
  start = as.POSIXct('01/01/1999 10:00', format = '%m/%d/%Y %H:%M'),
  end = as.POSIXct('01/01/2001 00:00', format = '%m/%d/%Y %H:%M'),
  units = 'days'
)

# Dates
calc_duration(
  start = as.Date('01/01/1999', format = '%m/%d/%Y'),
  end = as.Date('01/01/2001', format = '%m/%d/%Y'),
  units = 'years'
)
```

paste_efs *Paste Event-Free-Survival*

Description

Returns a human-readable event-free-survival from a survfit object and a specified time point.

Usage

```
paste_efs(fit = NA, time = NA, percent.sign = TRUE, digits = 1)
```

Arguments

fit	Required. survival::Surv() object. The time-to-event model of interest.
time	Required. Numeric. Indicates duration of time. Units are whatever was used to create the time-to-event model.
percent.sign	Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
digits	Optional. Integer. Number of digits to round to. Defaults to 1.

Examples

```
library(survival)
fit <- survfit(Surv(time, status) ~ 1, data = diabetic)
paste_efs(fit, c(1, 3, 5))
```

paste_freq *Paste Frequency*

Description

Returns a human-readable frequency from count(able) data. Supports vectorized data (i.e. dplyr::mutate()).

Usage

```
paste_freq(count = NA, total = NA, percent.sign = TRUE, digits = 1)
```

Arguments

count	Required. Tibble, Column (logical), or Numeric. The numerator. Tibbles and columns are automatically tallied (nrow or sum(na.rm = TRUE)).
total	Required. Tibble, Column, or Numeric. The denominator. Tibbles and columns are automatically tallied (nrow or sum(na.rm = TRUE)).
percent.sign	Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
digits	Optional. Integer. Number of digits to round to. Defaults to 1.

Examples

```
library(tibble)

# Numeric
paste_freq(20, 100)

# Tibble
data_tibble <- tibble(column = c(1:100))
paste_freq(data_tibble[1:20,], data_tibble)
```

paste_mean	<i>Paste Mean</i>
------------	-------------------

Description

Returns a human-readable mean with standard deviation from numeric data.

Usage

```
paste_mean(col = NA, less.than.one = FALSE, digits = 1)
```

Arguments

col	Required. Vector/Column (numeric). Data to summarize.
less.than.one	Optional. Logical. Indicates a mean that rounds to 0 should be printed as <1. Defaults to FALSE (0).
digits	Optional. Integer. Number of digits to round to. Defaults to 1.

Examples

```
paste_mean(mtcars$mpg)
```

paste_median	<i>Paste Median</i>
--------------	---------------------

Description

Returns a human-readable median with inter-quartile range from numeric data.

Usage

```
paste_median(col = NA, less.than.one = FALSE, digits = 1)
```

Arguments

<code>col</code>	Required. Vector/Column (numeric). Data to summarize.
<code>less.than.one</code>	Optional. Logical. Indicates a median that rounds to 0 should be printed as <1. Defaults to FALSE (0).
<code>digits</code>	Optional. Integer. Number of digits to round to. Defaults to 1.

Examples

```
paste_median(mtcars$mpg)
```

<code>tabulate_at_risk</code>	<i>Tabulate At Risk</i>
-------------------------------	-------------------------

Description

Returns a risk table from a model object and specified time points.

Usage

```
tabulate_at_risk(fit = NULL, times = NULL)
```

Arguments

<code>fit</code>	Required. <code>survival::survfit()</code> object.
<code>times</code>	Required. Numeric. One or vector of times to calculate for.

Value

Tibble risk table.

Examples

```
library(survival)
fit <- survfit(Surv(time, status) ~ 1, data = diabetic)
tabulate_at_risk(fit, c(1, 3, 5))
```

tabulate_model	<i>Tabulate Model</i>
----------------	-----------------------

Description

Converts parameters from a model object into a usable table for publication purposes. By default, formats the table into a human-readable/exportable form.

Usage

```
tabulate_model(fit, format, percent.sign, digits, p.digits)
```

Arguments

<code>fit</code>	Required. Model object. See S3 methods below.
<code>format</code>	Optional. Logical. Rounds numbers and formats text for a cleaner, readable output. Defaults to TRUE.
<code>percent.sign</code>	Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
<code>digits</code>	Optional. Integer. Number of digits to round to. Defaults to 1.
<code>p.digits</code>	Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded using 'digits'. Defaults to 4.

Value

Returns tibble containing summarizing statistics and tests.

See Also

[tabulate_model.lm](#) [tabulate_model.coxph](#) [tabulate_model.glm](#)

tabulate_model.coxph	<i>Tabulate Model: Cox PH</i>
----------------------	-------------------------------

Description

Converts parameters from a cox parametric hazards model into a usable table for publication purposes.

Usage

```
## S3 method for class 'coxph'
tabulate_model(fit, format = TRUE, percent.sign = TRUE,
  digits = 1, p.digits = 4)
```

Arguments

fit	Required. survival::coxph() object.
format	Optional. Logical. Rounds numbers and formats text for a cleaner, readable output. Defaults to TRUE.
percent.sign	Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
digits	Optional. Integer. Number of digits to round to. Defaults to 1.
p.digits	Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded using 'digits'. Defaults to 4.

See Also

[tabulate_model](#)

Examples

```
library(tibble)
library(survival)

# coxph() Object
coxph_data <- as_tibble(cgd)

tabulate_model(
  fit = coxph(
    Surv(tstart, tstop, status) ~ age + center + sex,
    data = coxph_data
  )
)
```

tabulate_model.glm *Tabulate Model: GLM*

Description

Converts parameters from a generalized linear model into a usable table for publication purposes.

Usage

```
## S3 method for class 'glm'
tabulate_model(fit, format = TRUE, percent.sign = TRUE,
  digits = 1, p.digits = 4)
```

Arguments

fit	Required. MASS::glm(family = 'binomial') object.
format	Optional. Logical. Rounds numbers and formats text for a cleaner, readable output. Defaults to TRUE.
percent.sign	Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
digits	Optional. Integer. Number of digits to round to. Defaults to 1.
p.digits	Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded using 'digits'. Defaults to 4.

See Also

[tabulate_model](#)

Examples

```
library(dplyr)
library(MASS)

# glm() Object
logit_data <- MASS::birthwt %>%
  mutate_at(c('race'), as.factor) %>%
  mutate_at(c('low', 'smoke', 'ht', 'ui'), as.logical)
tabulate_model(
  fit = stats::glm(
    low ~ race + smoke + age,
    data = logit_data,
    family = 'binomial'
  )
)
```

tabulate_model.lm *Tabulate Model: Linear Regression (LM)*

Description

Converts parameters from a linear regression model into a usable table for publication purposes.

Usage

```
## S3 method for class 'lm'
tabulate_model(fit, format = TRUE, percent.sign = TRUE,
  digits = 1, p.digits = 4)
```


Arguments

fit	Required. lm() object.
format	Optional. Logical. Rounds numbers and formats text for a cleaner, readable output. Defaults to TRUE.
percent.sign	Optional. Logical. Indicates percent sign should be printed for frequencies. Defaults to TRUE.
digits	Optional. Integer. Number of digits to round to. Defaults to 1.
p.digits	Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded using 'digits'. Defaults to 4.

See Also

[tabulate_model](#)

Examples

```
library(dplyr)

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

tabulate_model(fit = lm(mpg ~ vs + drat + cyl, data = data_mtcars))
```

test_factor

Test Factor Data

Description

Returns p-value from parametric or non-parametric testing of stratified categorical (factor) data.

Usage

```
test_factor(col = NULL, by = ".by", data = NULL, parametric = TRUE,
  digits = 1, p.digits = 4)
```

Arguments

col	Required. Character. Name of logical or factor column containing observations.
by	Required. Character. Name of logical or factor column to stratify by.
data	Required. Tibble. Data being used.
parametric	Optional. Logical. Indicates parametric testing should be used. (Chisquared). Defaults to FALSE (non-parametric; Fisher's Exact).
digits	Optional. Integer. Number of digits to round to. Defaults to 1.
p.digits	Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded using 'digits'. Defaults to 4.

test_numeric	<i>Test Numeric Data</i>
--------------	--------------------------

Description

Returns p-value from parametric or non-parametric testing of stratified continuous (numeric) data.

Usage

```
test_numeric(col = NULL, by = ".by", data = NULL,  
            parametric = FALSE, digits = 1, p.digits = 4)
```

Arguments

col	Required. Character. Name of numeric column containing observations.
by	Required. Character. Name of logical or factor column to stratify by.
data	Required. Tibble. Data being used.
parametric	Optional. Logical. Indicates parametric testing should be used. (Student's T-Test). Defaults to FALSE (non-parametric; Wilcox).
digits	Optional. Integer. Number of digits to round to. Defaults to 1.
p.digits	Optional. Integer. Number of p-value digits to print. Note that p-values are still rounded using 'digits'. Defaults to 4.

Index

`calc_duration`, 2

`paste_efs`, 3

`paste_freq`, 3

`paste_mean`, 4

`paste_median`, 4

`tabulate_at_risk`, 5

`tabulate_model`, 6, 7–9

`tabulate_model.coxph`, 6, 6

`tabulate_model.glm`, 6, 7

`tabulate_model.lm`, 6, 8

`test_factor`, 9

`test_numeric`, 10