

Package ‘tableeasy’

July 22, 2025

Title Tables of Clinical Study

Version 1.1.2

Description Creates some tables of clinical study. 'Table 1' is created by table1() to describe base-line characteristics, which is essential in every clinical study. Created by table2(), the function of 'Table 2' is to explore influence factors. And 'Table 3' created by table3() is able to make stratified analysis.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

Imports survival, stats, lmtest, mgcv, nortest, tableone

NeedsCompilation no

Author Jian Hang Zheng [aut, cre],
Yong Qi Chen [aut]

Maintainer Jian Hang Zheng <1040854241@qq.com>

Repository CRAN

Date/Publication 2022-04-08 02:40:02 UTC

Contents

div_custom	2
div_quantile	2
ensure_factor	3
smooth_curve	4
table1	6
table2	7
table3	9

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

div_custom	<i>Split a Variable by Custom Values</i>
------------	--

Description

Split a continuous variable by custom values. Converts a continuous variable to a categorical variable.

Usage

```
div_custom(var, div, data)
```

Arguments

var	A string. A variable to be summarized given as a string.
div	A numeric vector. The variable can be split into at least two levels by custom values.
data	A data frame in which these variables exist.

Value

A factor variable.

Examples

```
## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
library(tableeasay)
data(pbc)
## Check variables
head(pbc)
##
div_custom(var = 'age', div = c(40, 60), data = pbc)
```

div_quantile	<i>Split a Variable by Quantile Statistics</i>
--------------	--

Description

Split a continuous variable by quantile statistics. Converts a continuous variable to a categorical variable.

Usage

```
div_quantile(var, div, data)
```

Arguments

var	A string. A variable to be summarized given as a string.
div	A positive integer greater than 1 or a vector of integers. If a positive integer greater than 1, it is the number of factor levels when the variable is split by quantile statistics. If a vector of integers, it is the strategy of grouping x by quantile statistics and then merging groups.
data	A data frame in which these variables exist.

Value

A factor variable.

Examples

```
## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
library(tableeasay)
data(pbc)
## Check variables
head(pbc)
##
div_quantile(var = 'age', div = 5, data = pbc)
div_quantile(var = 'age', div = c(2,3), data = pbc)
```

ensure_factor

Ensure Factor Variable in Data Set

Description

Ensure that factor variables in data set are of the correct type.

Usage

```
ensure_factor(data, execute = FALSE, threshold_factor = 5)
```

Arguments

data	A data frame.
execute	Bool, default = FALSE. If TRUE, then output data frame.
threshold_factor	An integer, default = 5. Criteria to judge as factor type. When a variable has more than threshold_factor unique values, it is considered continuous.

Value

A list containing data frame and description.

Examples

```
## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
library(tableeasy)
data(pbc)
## Check variables
head(pbc)
##
ensure_factor(pbc)

ensure_factor(pbc,execute=TRUE)[['message']]
pbc_exe <- ensure_factor(pbc,execute=TRUE)[['data']]
```

smooth_curve

Smooth Curve

Description

Draw smooth curves. The four regression methods include general linear regression, logistic regression, conditional logistic regression and cox proportional hazards regression.

Usage

```
smooth_curve(
  x,
  y,
  data,
  y_time = NULL,
  strata = NULL,
  adj = c(),
  fx = FALSE,
  k = c(),
  split_var = NULL,
  div = c()
)
```

Arguments

x	A string. The independent variable to be summarized given as a string.
y	A string. The dependent variable to be summarized given as a string.
data	A data frame in which these variables exist.
y_time	A string. The survival time variable to be summarized given as a string.
strata	A string. The paired variable to be summarized given as a string.
adj	A vector of strings, default = c(). Moderator variables to be summarized given as a character vector.
fx	Bool, default = FALSE.

k	A vector of integers, default = c(). Degree of freedom, It only works when fx = TRUE.
split_var	A string, default = NULL. The split variables to be summarized given as a string.
div	A numeric vector, default = c(). It only works when split_var is a continuous variable.

Value

An object about smooth curve.

Examples

```
## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
library(tableeasy)
data(pbc)
## Check variables
head(pbc)
##The censored data is not discussed here
pbc_full <- subset(pbc,status!=0)
pbc_full$status <- pbc_full$status-1
## Make categorical variables factors
varsToFactor <- c('status','trt','ascites','hepato','spiders','edema','stage','sex')
pbc_full[varsToFactor] <- lapply(pbc_full[varsToFactor], factor)

## Moderator variables
adj_pbc <- c('age','alk.phos','ast')

## Smooth curve of General linear regression:
gam <- smooth_curve(x='albumin',
                    y='bili',
                    adj=adj_pbc,
                    data=pbc_full)
plot(gam$gam,se=TRUE,rug=TRUE,shift=gam$shift)

## Smooth curve of logistic regression:
gam <- smooth_curve(x = 'albumin',
                    y = 'status',
                    adj = adj_pbc,
                    split_var = 'age',
                    div = c(45),
                    data = pbc_full)
plot(gam$gam[[1]],se=FALSE,rug=TRUE,xlim=c(2,4.5),ylab = 'Adjusted ln ORs for death')
oldpar <- par(new=TRUE)
plot(gam$gam[[2]],se=FALSE,rug=TRUE,xlim=c(2,4.5),ylab = 'Adjusted ln ORs for death',lty=2)
on.exit(par(oldpar))

## Smooth curve of conditional logistic regression:
pbc_full <- data.frame(pbc_full,'ytime'=1)
gam <- smooth_curve(x = 'albumin',
                    y_time = 'ytime',
```

```

      y = 'status',
      adj = adj_pbc,
      strata = 'trt',
      data = pbc_full)

termplot(gam,term =c(1),col.term ="black",col.se = "black",se=TRUE,rug=FALSE,
         ylab="Log ORs for death")

## Smooth curve of Cox proportional hazards regression:
gam <- smooth_curve(x ='albumin',
                    y_time = 'time',
                    y = 'status',
                    adj = adj_pbc,
                    data = pbc_full)
termplot(gam,term =c(1),col.term ="black",col.se = "black",se=TRUE,rug=FALSE)

```

table1

Table 1

Description

Create an object summarizing all baseline variables (both continuous and categorical). The function is improved on the basis of `tableone::CreateTableOne` to become more convenient to use.

Usage

```

table1(
  var,
  strata,
  data,
  normal = c("age", "bmi", "sbp", "dbp"),
  catDigits = 1,
  contDigits = 1,
  pDigits = 3,
  showAllLevels = FALSE
)

```

Arguments

<code>var</code>	A vector of strings. Variables to be summarized given as a character vector.
<code>strata</code>	A vector of strings. Stratifying (grouping) variable name(s) given as a character vector.
<code>data</code>	A data frame in which these variables exist.
<code>normal</code>	A vector of strings, default = <code>c("age", "bmi", "sbp", "dbp")</code> . Show the mean and standard deviation of some variables.
<code>catDigits</code>	An integer, Number of decimal places in the table of continuous variables.
<code>contDigits</code>	An integer, Number of decimal places in the table of categorical variables.

pDigits An integer, Number of decimal places in the table of p values.

showAllLevels Bool, default = FALSE. If TRUE, the table contains all levels of categorical variables.

Value

An object describing baseline characteristics.

Examples

```
## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
library(tableeasy)
data(pbc)
## Check variables
head(pbc)
## Make categorical variables factors
varsToFactor <- c('status','trt','ascites','hepato','spiders','edema','stage','sex')
pbc[varsToFactor] <- lapply(pbc[varsToFactor], factor)
##Table 1
table1(var=c('age','albumin','alk.phos','ast','edema','ascites','bili','chol'),strata='trt',pbc)
```

table2

Table 2

Description

' Table 2 ' was created through regression analysis to research influence factor. The four regression methods include general linear regression, logistic regression, conditional logistic regression and cox proportional hazards regression.

Usage

```
table2(
  x,
  y,
  data,
  y_time = NULL,
  strata = NULL,
  adj = c(),
  div = list(),
  div_num = list(),
  ref = c(),
  ref_num = c(),
  continuous = FALSE,
  case = 2,
  method = "general",
  outformat = 2
)
```

Arguments

<code>x</code>	A string. The independent variable to be summarized given as a string.
<code>y</code>	A string. The dependent variable to be summarized given as a string.
<code>data</code>	A data frame in which these variables exist.
<code>y_time</code>	A string. The survival time variable to be summarized given as a string. It only works when <code>method = "cox"</code> .
<code>strata</code>	A string. The paired variable to be summarized given as a string. It only works when <code>method = "con_logistic"</code> .
<code>adj</code>	A vector of strings, default = <code>c()</code> . Moderator variables to be summarized given as a character vector.
<code>div</code>	A list containing Positive int greater than 1 or integer vector, If a positive integer greater than 1, it is the number of factor levels when <code>x</code> is split by quantile statistics. If a vector of integers, it is the strategy of grouping <code>x</code> by quantile statistics and then merging groups.
<code>div_num</code>	A list containing numeric vectors, Elements in the list are custom values, and <code>x</code> can be split into at least two levels by elements in the list.
<code>ref</code>	A vector of integers. The control level of factor levels when <code>x</code> is split by quantile statistics.
<code>ref_num</code>	A vector of integers. The control level of factor levels when <code>x</code> is split by custom values.
<code>continuous</code>	Bool, default = FALSE. If default, then does not output the result when the independent variable is regarded as a continuous variable.
<code>case</code>	A vector of integers, default = 2. The case level of <code>y</code> .
<code>method</code>	("general", "logistic", "con_logistic", "cox"), default = "general".
<code>outformat</code>	1 or 2, default = 2, Output format. It only works when <code>method = "general"</code> . The table output mean(sd) when <code>outformat=1</code> and output median(IQR) when <code>outformat=2</code> .

Value

An object researching influence factor.

Examples

```
## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
library(tableeasy)
data(pbc)
## Check variables
head(pbc)
##The censored data is not discussed here
pbc_full <- subset(pbc,status!=0)
pbc_full$status <- pbc_full$status-1
## Make categorical variables factors
varsToFactor <- c('status','trt','ascites','hepato','spiders','edema','stage','sex')
```

```

pbc_full[varsToFactor] <- lapply(pbc_full[varsToFactor], factor)

## Moderator variables
adj_pbc <- c('age', 'alk.phos', 'ast')

## General linear regression:
table2(x = 'albumin', y = 'bili',
      adj = c(), data = pbc_full,
      div = list(5, c(2, 3)), div_num = list(c(3.2, 4)),
      ref = c(2, 1), ref_num = c(2),
      outformat = 2)

## Logistic regression:
table2(x = 'albumin', y = 'status',
      adj = adj_pbc, data = pbc_full,
      div = list(5, c(2, 3)),
      method = 'logistic')

## Conditional logistic regression:
table2(x = 'albumin', y = 'status', strata = 'trt',
      adj = adj_pbc, data = pbc_full,
      div = list(5, c(2, 3)),
      method = 'con_logistic')

## Cox proportional hazards regression:
table2(x = 'albumin', y = 'status', y_time = 'time',
      adj = adj_pbc, data = pbc_full,
      div = list(5, c(2, 3)),
      method = 'cox')

```

table3

Table 3

Description

Creates 'Table 3' which is about stratified analysis. The three regression methods include general linear regression, logistic regression and cox proportional hazards regression.

Usage

```

table3(
  x,
  y,
  data,
  split_var,
  y_time = NULL,
  adj = c(),
  split_div = list(),
  outformat = 4,
  method = "general"
)

```

Arguments

<code>x</code>	A string. The independent variable to be summarized given as a string.
<code>y</code>	A string. The dependent variable to be summarized given as a string.
<code>data</code>	A data frame in which these variables exist.
<code>split_var</code>	A vector of strings. Strata variables to be summarized given as a character vector.
<code>y_time</code>	A string. The survival time variable to be summarized given as a string. It only works when <code>method = "cox"</code> .
<code>adj</code>	A vector of strings, default = <code>c()</code> . Moderator variables to be summarized given as a character vector.
<code>split_div</code>	A list containing numeric vectors or a vector of integers that are summarized given as a string, default = <code>list()</code> . If default, it represents the strata variables are split by median or it is itself a categorical variable. If the element of list is a numeric vector, it represents the strata variables are split by custom values. And if the element of list is a vector of integers that are summarized given as a string, it represents the strata variables are split by quantile statistics.
<code>outformat</code>	1 or 2 or 3 or 4, default = 4. Output format.
<code>method</code>	("general", "logistic", "cox"), default = "general".

Value

An object about stratified analysis.

Examples

```
## Load Mayo Clinic Primary Biliary Cirrhosis Data
library(survival)
library(tableeasy)
data(pbc)
## Check variables
head(pbc)
##The censored data is not discussed here
pbc_full <- subset(pbc,status!=0)
pbc_full$status <- pbc_full$status-1
## Make categorical variables factors
varsToFactor <- c('status','trt','ascites','hepato','spiders','edema','stage','sex')
pbc_full[varsToFactor] <- lapply(pbc_full[varsToFactor], factor)

## Moderator variables
adj_pbc <- c('age','alk.phos','ast')
## Converts the continuous variables named 'albumin' to a categorical variable named 'albumin_2'.
albumin_2 <- div_quantile('albumin',div = c(2),pbc_full)
pbc_full <- data.frame(pbc_full,'albumin_2' = albumin_2)

## General linear regression:
table3(x = 'albumin_2', y = 'bili',
      adj = c(), data = pbc_full,
      split_var = c('age','alk.phos','ast','trt'), split_div = list(),
```

```
        outformat = 1)

## Logistic regression:
table3(x = 'albumin_2', y = 'status',
      adj = adj_pbc, data = pbc_full,
      split_var = c('age', 'alk.phos', 'ast', 'trt'), split_div = list(c('2', '3'), c('3')),
      outformat = 2, method = 'logistic')

## Cox proportional hazards regression:
table3(x = 'albumin_2', y = 'status', y_time = 'time',
      adj = adj_pbc, data = pbc_full,
      split_var = c('age', 'alk.phos', 'ast', 'trt'), split_div = list(c(45), c(1500, 1700), c(), c()),
      outformat = 3, method = 'cox')
```

Index

div_custom, [2](#)
div_quantile, [2](#)

ensure_factor, [3](#)

smooth_curve, [4](#)

table1, [6](#)
table2, [7](#)
table3, [9](#)