Package ‘fastStat’

September 17, 2020

Title Faster for Statistic Work

Version 1.4

Description When we do statistic work, we need to see the structure of the data. list.str() function will help you see the structure of the data quickly. list.plot() function can help you check every variable in your dataframe. table_one() function will make it easy to make a baseline table including difference tests. uv_linear(), uv_logit(), uv_cox(), uv_logrank() will give you a hand to do univariable regression analysis, while mv_linear(), mv_logit() and mv_cox() will carry out multivariable regression analysis.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports set, reshape2, do, plyr, car, e1071, tseries, survival, ggplot2, ggrepel, PredictABEL, survIDINRI

URL https://github.com/yikeshu0611/fastStat

BugReports https://github.com/yikeshu0611/fastStat/issues

Suggests rms

Depends survC1

NeedsCompilation no

Author Jing Zhang [aut, cre],
Zhi Jin [aut]

Maintainer Jing Zhang <zj391120@163.com>

Repository CRAN

Date/Publication 2020-09-17 15:00:13 UTC

R topics documented:

cor2 *Correlation Analysis*

Description

Correlation Analysis

Usage

```
cor2(data, x1, x2, method = "spearman")
```

Arguments

`data`	dataframe
`x1`	x1
`x2`	x2
`method`	1,2,3

`cor_sig` *Correlation Analysis with Signicant Values*

Description

Correlation Analysis with Signicant Values

Usage

```
cor_sig(data, method = "pearson")
```

Arguments

`data`	a dataframe or matrix
`method`	a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated.

Value

correlation analysis with significant star.

Examples

```
cor_sig(mtcars)
```

`cor_sig_star` *Correlation Analysis with Sinificant and Correlation Value*

Description

Correlation Analysis with Sinificant and Correlation Value

Usage

```
cor_sig_star(data, method = "pearson")
```

Arguments

`data`	a dataframe or matrix
`method`	a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default, 1), "kendall" (2), or "spearman" (3): can be abbreviated.

Value

correlation analysis with significant star.

Examples

```
cor_sig_star(mtcars)
```

`cor_star` *Correlation Analysis*

Description

Correlation Analysis

Usage

```
cor_star(data, method = "pearson")
```

Arguments

`data`	a dataframe or matrix
`method`	a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default, 1), "kendall" (2), or "spearman" (3): can be abbreviated.

Value

correlation analysis with significant star.

Examples

```
cor_star(mtcars)
```

`digital` *Set Digital Number*

Description

Set Digital Number

Usage

```
digital(x, round)
```

Arguments

`x`	vector, dataframe or matrix
`round`	digital number

Value

character with the same digital number

Examples

```
digital(1.2,4)
```

`idi` *Perform IDI for logistic and cox regression*

Description

Easy to perform integrated discrimination improvement index (IDI) for logistic and cox regression.

Usage

```
idi(f1, f2, ...)

## S3 method for class 'lrm'
idi(f1, f2, ...)

## S3 method for class 'cph'
idi(f1, f2, timepoint = NULL, ...)

## S3 method for class 'coxph'
idi(f1, f2, timepoint = NULL, ...)
```

Arguments

`f1`	base model
`f2`	the other model
`...`	ignore
`timepoint`	one time point for cox regression, default is median time.

Value

IDI results

Examples

```
library(rms)
# logistic

data(lung)

lung$status=lung$status-1
f1=lrm(status~age+sex,lung)
f2=lrm(status~age+sex+ph.ecog,lung)
idi(f1,f2)

# survival

head(lung)

range(lung$time)

lung=lung[complete.cases(lung),]

f1=cph(Surv(time,status)~age+sex,lung)
f2=cph(Surv(time,status)~age+sex+ph.ecog,lung)
idi(f1,f2)
```

`list.factor` *Return All Factor Variables*

Description

Return all factor variables in a dataframe or matrix

Usage

```
list.factor(x, levels = FALSE)
```

Arguments

`x`	a dataframe or matrix
`levels`	logical. TRUE to display levels for factor variable.

Value

factor variable names and levels

Examples

```
jh=data.frame(x=c(1,2,3,1),
    k=c(4,5,6,7),
    h=c('a','a','b','b'))
list.factor(jh)
```

`list.NA` *Return Na Count and Percentage*

Description

Return Na count and percentage for each variable in a dataframe or matrix.

Usage

```
list.NA(x)
```

Arguments

`x`	a numeric vector, a dataframe or matix

Value

A dataframe contains NA variable names, NA count and percentage

Examples

```
 jh=data.frame(x=c(1,2,3,1),
     k=c(4,5,6,7),
     h=c('a','a',NA,'D'),
     f=c(1,2,NA,NA))
list.NA(jh)
```

`list.numeric` *Return All Numeric Variables in A Dataframe*

Description

Return All Numeric Variables in A Dataframe

Usage

```
list.numeric(df)
```

Arguments

`df`	a dataframe

Value

numeric variable names

Examples

```
jh=data.frame(x=c(1,2,3,1),
    k=c(4,5,6,7),
    h=c('a','a','b','b'))
list.numeric(jh)
```

`list.plot` *Scatter Plot for Single Value*

Description

Scatter Plot for Single Value

Usage

```
list.plot(x, label = "x")
```

Arguments

`x`	vector, dataframe or matrix
`label`	labels for points. If label equals x, defaulted, id will be added. If label equals y, y value will be added. If label equals xy, id and y value will be added.

Value

sactter

Examples

```
    list.plot(mtcars)
```

`list.str` *Structure for Data*

Description

Structure for Data

Usage

```
list.str(x, n = 3)
```

Arguments

`x`	a dataframe or matrix
`n`	the maximum level number to display

Value

a dataframe contains variable names and class

Examples

```
jh=data.frame(x=c(1,2,3,1),
              k=c(4,5,6,7),
              h=c('a','a','b','b'))
list.str(x = jh)
```

`list.summary` *Summary for Data*

Description

Summary for Data

Usage

```
list.summary(x, round = 2)
```

Arguments

x	numeric
round	digital number

Value

a dataframe with min, max, quantile 25 and 75, mean, median, sd and NA

Examples

```
list.summary(mtcars)
```

mv_cox *Multivariable Logistic Regression*

Description

Multivariable Logistic Regression

Usage

```
mv_cox(data, time, event, x, direction = "no", summary = TRUE, ...)
```

Arguments

data	data
time	time variable
event	event variable
x	variable names for univariable logistic regression. If missing, it will be column names of data except y and adjust
direction	direction for stepwise regression. Four options: no, backward, forward and both. Defaulted is no
summary	logical. Whether to return summary results. TRUE as defaulted
...	arguments passed to step() function.

Value

multivariable logistic regression results

Examples

```
mv_cox(data = mtcars,
    time = 'qsec',event = 'am',
    direction = 'both')
```

mv_linear *Multivariable Linear Regression*

Description

Multivariable Linear Regression

Usage

```
mv_linear(data, y, x, direction = "no", summary = TRUE, ...)
```

Arguments

`data` data

`y` y varaible

`x` variable names for univariable linear regression. If missing, it will be column names of data except y and adjust

`direction` direction for stepwise regression. Four options: no, backward, forward and both. Defaulted is no

`summary` logical. Whether to return summary results. TRUE as defaulted

`...` arguments passed to step() function

Value

multivariable linear regression results

Examples

```
mv_linear(data = rock,y = 'perm',
    direction = 'both')
```

mv_logit *Multivariable Logistic Regression*

Description

Multivariable Logistic Regression

Usage

```
mv_logit(data, y, x, direction = "no", summary = TRUE, ...)
```

Arguments

`data` data

`y` y varaible

`x` variable names for univariable logistic regression. If missing, it will be column names of data except y and adjust

`direction` direction for stepwise regression. Four options: no, backward, forward and both. Defaulted is no

`summary` logical. Whether to return summary results. TRUE as defaulted

`...` arguments passed to step() function

Value

multivariable logistic regression results

Examples

```
mv_logit(data = mtcars,y = 'am',
    variable = c('cyl','disp'))
```

`normal` *Normal Distribution Test*

Description

Using Jarque Bera test, shapiro wilk test and Kolmogorov Smirnov test for one numeric object or numeric object in dataframe or matrix. Na is omitted in each object.

Usage

```
normal(x, num.names)
```

Arguments

`x` numerica object or dataframe and matrix

`num.names` numeric column names for dataframe and matrix. If missing, all numeric column names will be given.

Value

a dataframe containing kurtosis, skewness and p value for Jarque Bera test, shapiro wilk test and Kolmogorov Smirnov test. In star column, star represents p > 0.05, while underline taking the opposite.

Examples

```
set.seed(2019)
rn1=rnorm(100,0,2)
df=data.frame(rn1=rnorm(100,0,2),
              rn2=rnorm(100,2,4))
#normal test for one object
normal(rn1)

#normal test for dataframe
normal(df)
```

`round<-` *Change the Digital for Double*

Description

Change the Digital for Double

Usage

```
round(x) <- value
```

Arguments

`x`	a double number
`value`	digital number

Value

double number

Examples

```
x = 3.123
#usual method
x = round(x, 3)
#now
round(x) = 3
```

survdiff_p.value *Extract P Value after survdiff() function*

Description

Extract P Value after survdiff() function

Usage

```
survdiff_p.value(survdiff)
```

Arguments

survdiff	the results of survdiff() function

Value

p value

Examples

```
library(survival)
diff_result=survdiff(Surv(qsec,vs)~cyl,data=mtcars)
survdiff_p.value(diff_result)
```

survSum *Calculate Survival Rate and Time*

Description

Calculate Survival Rate and Time

Usage

```
surv_table(fit)

surv_median_time(fit)

surv_year_rate(fit, year)
```

Arguments

fit	fit by survfit() function
year	year

Value

a dataframe

Examples

```
library(survival)
fit=survfit(Surv(futime, fustat)~rx, data=ovarian)

#survival table
surv_table(fit)

#median survival rate
surv_median_time(fit)

#one year survaival rate
surv_year_rate(fit,365)

#two years survaival rate
surv_year_rate(fit,365*2)
```

table_one *Get Summary Table*

Description

Get the first summary table when study.

Usage

```
table_one(
  data,
  group,
  mean_sd,
  median_q4,
  median_range,
  count_percent,
  mean,
  median,
  max,
  min,
  sd,
  q25,
  q75,
  count,
  percent,
  round = 2,
  count.percent.direction = "v",
  t.test,
  anova,
  wilcox.test,
  kruskal.test,
  chisq.test,
  fisher.test,
  weighted,
  statistics = FALSE
)
```

Arguments

`data`	data that will be summarized
`group`	one or more group variable names
`mean_sd`	variable names for mand and standard deviation. in the results represents plus and minus
`median_q4`	variable names for median and 25 and 75 quantiles
`median_range`	variable names for median and range
`count_percent`	variable names for count and percentage
`mean`	variable names for mean
`median`	variable names for median
`max`	variable names for max
`min`	variable names for min
`sd`	variable names for standard deviation
`q25`	variable names for 25 quantile
`q75`	variable names for 75 quantile
`count`	variable names for count
`percent`	variable names for percentage
`round`	digital round. 2 is defaulted
`count.percent.direction`	calculate of direction for count, percent and count_percent arguments, which should be one of g, group, v or var, v as defaulted
`t.test`	two-side t test
`anova`	two-side anova
`wilcox.test`	two-side wilcox test
`kruskal.test`	two-side kruskal test
`chisq.test`	two-side chisq test
`fisher.test`	two-side fisher test
`weighted`	weight for data
`statistics`	a logical object. TRUE to display the statistic information. Default is FALSE

Value

a summary matrix

Examples

```
table_one(data = mtcars,group='vs',
          mean_sd = 'wt',
          count_percent  = c('gear','am')
)

table_one(data = mtcars,
group='vs',

mean_sd = 'wt',
t.test = 'wt',

count_percent  = c('gear','am','cyl'),
chisq.test = c('am','gear'),
fisher.test = c('cyl'),

round = 3
)
```

`to.factor` *Set Factor Class*

Description

Set Factor Class

Usage

```
to.factor(x, levels)
```

Arguments

`x`	the data that you want to set
`levels`	levels, the first levels is the reference. If the length of levels is 1, no levels will be given to x

Value

factor x

Examples

```
to.factor(mtcars$gear,c(4,3,5))
to.factor(mtcars$gear)
```

to.factor<- *Set Factor Class*

Description

Set Factor Class

Usage

```
to.factor(x) <- value
```

Arguments

x	the data that you want to set
value	levels, the first value is the reference. If the length of value is 1, no levels will be given to x

Value

factor x

Examples

```
to.factor(mtcars$gear) <- c(4,3,5)
```

to.labels *Give Labels to Factor*

Description

Give Labels to Factor

Usage

```
to.labels(x, labels)
```

Arguments

x	factor or numeric variable
labels	labels separated by colon

Value

factor variable with lables, the first lable will be treated as reference.

Examples

```
to.labels(x=mtcars$am,labels=c('0:Female','1:Man'))
```

`to.labels<-` *Give Labels to Factor*

Description

Give Labels to Factor

Usage

```
to.labels(x) <- value
```

Arguments

x	factor or numeric variable
value	labels separated by colon

Value

factor variable with lables, the first lable will be treated as reference.

Examples

```
to.labels(x=mtcars$am) <- c('0:Female','1:Man')
```

`to.numeric` *Change to Numeric Form*

Description

Change to Numeric Form

Usage

```
to.numeric(x)
```

Arguments

x	vector

Value

numeric data

Examples

```
x=c(1,2,3)
to.factor(x) <- 1
to.numeric(x)
```

to.numeric<- *Change to Numeric Form*

Description

Change to Numeric Form

Usage

```
to.numeric(x) <- value
```

Arguments

x	vector
value	anything, which will be ignored

Value

numeric data

Examples

```
x=c(1,2,3)
to.factor(x) <- 1
to.numeric(x) <- 1
```

to.refer *Set Refer for Factor*

Description

Convert data to be factor and set reference.

Usage

```
to.refer(x, refer)
```

Arguments

x	the data that you want to set
refer	refering level

Value

refered factor refer

Examples

```
to.refer(mtcars$vs,1)
```

`to.refer<-` *Set Refer for Factor*

Description

Convert data to be factor and set reference.

Usage

```
to.refer(x) <- value
```

Arguments

`x`	the data that you want to set
`value`	refering level

Value

refered factor value

Examples

```
to.refer(mtcars$vs) = 1
```

`uv_cox` *Looping for Univariable Cox Regression*

Description

Looping for Univariable Cox Regression

Usage

```
uv_cox(
  data,
  time,
  event,
  variable,
  adjust,
  round = 3,
  p_threshold = 0.05,
  order_by.hr = TRUE
)
```

Arguments

`data`	data
`time`	time variable
`event`	event variable
`variable`	variable names for univariable cox regression. If missing, it will be column names of data except y and adjust
`adjust`	adjust variable names for univariable cox regression
`round`	digital round, 3 is defaulted
`p_threshold`	threshold for p value to show star. 0.05 is defaulted
`order_by.hr`	logical. TRUE means order in or by decreasing. FLASE is defaulted

Value

univariable cox regression results

Examples

```
uv_cox(data = mtcars,
    time = 'qsec',event = 'vs')
```

`uv_linear` *Looping for Univariable Logistic Regression*

Description

Looping for Univariable Logistic Regression

Usage

```
uv_linear(
  data,
  y,
  variable,
  adjust,
  round = 3,
  p_threshold = 0.05,
  order_by.beta = TRUE
)
```

Arguments

`data`	data
`y`	y
`variable`	variable names for univariable logistic regression. If missing, it will be column names of data except y and adjust
`adjust`	adjust variable names for univariable logistic regression
`round`	digital round, 3 is defaulted
`p_threshold`	threshold for p value to show star. 0.05 is defaulted
`order_by.beta`	logical. TRUE means order in or by decreasing. FLASE is defaulted

Value

univariable logistic regression results

Examples

```
uv_linear(data = mtcars,y = 'vs')
```

`uv_logit` *Looping for Univariable Logistic Regression*

Description

Looping for Univariable Logistic Regression

Usage

```
uv_logit(
  data,
  y,
  variable,
  adjust,
  round = 3,
  p_threshold = 0.05,
  order_by.or = TRUE
)
```

Arguments

`data`	data
`y`	y
`variable`	variable names for univariable logistic regression. If missing, it will be column names of data except y and adjust
`adjust`	adjust variable names for univariable logistic regression
`round`	digital round, 3 is defaulted
`p_threshold`	threshold for p value to show star. 0.05 is defaulted
`order_by.or`	logical. TRUE means order in or by decreasing. FLASE is defaulted

Value

univariable logistic regression results

Examples

```
uv_logit(data = mtcars,y = 'vs')
```

---

uv_logrank	*Looping for logrank Regression*

Description

Looping for logrank Regression

Usage

```
uv_logrank(data, time, event, variable, round = 3, order_by.p = TRUE)
```

Arguments

`data`	data
`time`	time variable
`event`	event variable
`variable`	variable names for logrank regression. If missing, it will be column names of data except y
`round`	digital round, 3 is defaulted
`order_by.p`	logical. TRUE, defaulted, means increasing order in p value

Value

logrank regression results

Examples

```
uv_logrank(data = mtcars,
    time = 'qsec',event = 'vs')
```

Index