Package ‘moonBook’

July 23, 2025

Title Functions and Datasets for the Book by Keon-Woong Moon

Version 0.3.1

URL `https://github.com/cardiomoon/moonBook`

BugReports `https://github.com/cardiomoon/moonBook/issues`

Description Several analysis-related functions for the book entitled ``R statistics and graph for medical articles'' (written in Korean), version 1, by Keon-Woong Moon with Korean demographic data with several plot functions.

Depends R (>= 3.1.2)

License GPL-2

LazyData true

Encoding UTF-8

Imports nortest, survival, sjmisc, stringr, magrittr, crayon

Suggests testthat, knitr, ggplot2, sjlabelled, ztable, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.1.2

NeedsCompilation no

Author Keon-Woong Moon [aut, cre]

Maintainer Keon-Woong Moon <`cardiomoon@gmail.com`>

Repository CRAN

Date/Publication 2022-01-05 07:40:02 UTC

Contents

`acs` *Demographic data of 857 patients with ACS*

Description

A dataset containing demographic data and laboratory data of 857 patients with acute coronary syndrome(ACS).

Format

A data frame with 857 rows and 17 variables:

age	patient age in years
sex	"Male" or "Female"
cardiogenicShock	"No" or "Yes"
entry	vascular access route, either "Femoral" or "Radial"
Dx	Final diagnosis, One of the followings : STEMI, NSTEMI or Unstable Angina
EF	ejection fraction, percentage by echocardiography
height	height in centimeter
weight	weight in kilogram
BMI	body mass index in kg/m2
obesity	obesity, "No" or "Yes"
TC	total cholesterol level in mg/dL
LDLC	low density lipoprotein cholesterol level in mg/dL
HDLC	high density lipoprotein cholesterol level in mg/dL
TG	triglyceride level in mg/dL
DM	history of diabetes mellitus,"No" or "Yes"
HBP	history of hypertension,"No" or "Yes"
smoking	history of smoking, One of the followings : "Never","Ex-smoker","Smoker"

`addComma` *Change numbers into formatted numbers*

Description

Change numbers into formatted numbers

Usage

```
addComma(x)

## S3 method for class 'mytable'
addComma(x)

## S3 method for class 'mytable.df'
addComma(x)

## S3 method for class 'cbind.mytable'
addComma(x)

## S3 method for class 'data.frame'
addComma(x)

## S3 method for class 'character'
addComma(x)
```

Arguments

`x`	An object

Methods (by class)

- `mytable`: S3 method for class mytable
- `mytable.df`: S3 method for class mytable.df
- `cbind.mytable`: S3 method for class cbind.mytable
- `data.frame`: S3 method for class data.frame
- `character`: S3 method for class character

Examples

```
## Not run:
require(stringr)
require(magrittr)
require(ggplot2)
mytable(cut~.,data=diamonds) %>% addComma
x=mytable(Dx~sex,data=acs)
addComma(x)

## End(Not run)
```

`addLabelDf` *Add value labels to the data.frame*

Description

Add value labels to the data.frame

Usage

```
addLabelDf(data, mapping = NULL)
```

Arguments

`data`	A data.frame
`mapping`	Set of aesthetic mappings created by aes or aes_.

`cat.test` *Perform chisq.test or fisher test*

Description

Perform chisq.test or fisher test

Usage

```
cat.test(x, mode = 1, ...)
```

Arguments

`x`	a numeric vector or matrix. x and y can also both be factors.
`mode`	An integer. If 1(default), perform chisq.test first, If 2, perform fisher.test first
`...`	Further arguments to be passed to chisq.test or fisher.test

---

`cbind.mytable` *cbind function for class "mytable"*

Description

cbind function for class "mytable"

Usage

```
## S3 method for class 'mytable'
cbind(..., caption, y = NULL)
```

Arguments

`...`	Objects of class "mytable", a result of a call to `mytable`
`caption`	Unique values of grouping variables used for column name of table
`y`	Names of grouping variables used for caption of table

`centerprint` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
centerprint(x, ..., width = 10)
```

Arguments

`x`	a character vector
`...`	further arguments passed to or from other methods.
`width`	an integer

`changeColnameLabel` *Change column names with labels*

Description

Change column names with labels

Usage

```
changeColnameLabel(data)
```

Arguments

`data`	A data.frame

`comma` *Convert number to formatted number*

Description

Convert number to formatted number

Usage

```
comma(x, ...)
```

Arguments

`x`	A numeric vector
`...`	Further arguments to be passed to function format

`compress` *Compress an object of class mytable or cbind.mytable*

Description

Compress an object of class mytable or cbind.mytable

Usage

```
compress(x, no = 2, add.label = TRUE)

## S3 method for class 'mytable'
compress(x, no = 2, add.label = TRUE)

## S3 method for class 'cbind.mytable'
compress(x, no = 2, add.label = TRUE)

## S3 method for class 'data.frame'
compress(x, no = 2, add.label = TRUE)
```

Arguments

`x`	An object of class mytable or cbind.mytable
`no`	Representative group of two groups
`add.label`	Logical. Whether or not add representative group name

Methods (by class)

- `mytable`: S3 method for class mytable
- `cbind.mytable`: S3 method for class cbind.mytable
- `data.frame`: S3 method for class data.frame

Examples

```
require(stringr)
require(magrittr)
mytable(acs) %>% compress
mytable(Dx~.,data=acs) %>% compress
## Not run:
require(ztable)
mytable(Dx~.,data=acs) %>% compress %>% ztable
mytable(Dx+sex~.,data=acs) %>% compress

## End(Not run)
```

`deleteRows` *Delete rows of class mytable object*

Description

Delete rows of class mytable object

Usage

```
deleteRows(x, rows)
```

Arguments

`x`	An object of class mytable or cbind.mytable
`rows`	rows to delete

`densityplot` *Make Kernel density plot*

Description

Make Kernel density plot

Usage

```
densityplot(formula, data, main = "", xlab = "", ylab = "", ...)
```

Arguments

`formula`	an R model formula, of the form ~ variable to estimate the unconditional density of variable, or variable ~ factor to estimate the density of variable within each level of factor.
`data`	an optional data frame containing the data.
`main`	main title of plot
`xlab`	label for the horizontal-axis; defaults to the name of the variable x.
`ylab`	label for the vertical axis; defaults to "Density".
`...`	arguments to be passed to plot

Value

This function return NULL invisibly and draw graphs.

Examples

```
require(moonBook)
data(acs)
densityplot(age~Dx,data=acs)
```

`extractHR` *Extract hazard ratio from a data.frame*

Description

Extract hazard ratio from a data.frame

Usage

```
extractHR(x, digits = 2)
```

Arguments

`x`	an object of class coxph
`digits`	An integer indicating the number of decimal places (round) or significant digits to be used. Default value is 2.

Value

a data.frame consist of hazard ratio and 95 the p values.

Examples

```
require(survival)
data(cancer)
fit=coxph(Surv(time,status)~age+sex+obstruct+perfor,data=colon)
extractHR(fit)
```

`extractKind` *Extract kind of an object of class mytable*

Description

Extract kind of an object of class mytable

Usage

```
extractKind(df)
```

Arguments

`df`	An object of class mytable or cbind.mytable

`extractOR` *Extract the odds ratios from a S3 object of glm*

Description

Extract the odds ratios from a S3 object of glm

Usage

```
extractOR(x, digits = 2, method = "default")
```

Arguments

`x`	A S3 object of glm
`digits`	An integer indicating the number of decimal places (round) or significant digits to be used. Default value is 2.
`method`	Method to compute confidence interval. Choices are one of c("default","LRT").

Value

A data.frame consist of odds ratios and 95 p values

Examples

```
data(cancer,package="survival")
x=glm(status~rx+sex+age+obstruct+nodes,data=colon,family="binomial")
extractOR(x)
```

`getLabel` *Add column labels to the data.frame*

Description

Add column labels to the data.frame

Usage

```
getLabel(data, colname, use.column.label = TRUE)
```

Arguments

`data`	A data.frame
`colname`	character. column name
`use.column.label`	Logical. Whether or not use column labels.

getMapping *extract variable name from mapping, aes*

Description

extract variable name from mapping, aes

Usage

```
getMapping(mapping, varname)
```

Arguments

`mapping`	aesthetic mapping
`varname`	variable name to extract

Value

variable name in character

Examples

```
require(ggplot2)
mapping=aes(colour=sex)
getMapping(mapping,"colour")
getMapping(mapping,"x")
```

HRplot *Draw a hazard ratio plot*

Description

Draw a hazard ratio plot

Usage

```
HRplot(
  out,
  type = 1,
  xlab = "",
  ylab = "",
  show.OR = TRUE,
  show.CI = FALSE,
  sig.level = 1,
  cex = 1.2,
  lwd = 2,
```

```
  pch = 18,
  col = NULL,
  ...
)
```

Arguments

`out`	an object of class coxph or a resultant data.frame of mycph function
`type`	an integer indicating the type of plot. Default value is 1
`xlab`	a title for the x axis
`ylab`	a title for the y axis
`show.OR`	a logical vector indicating whether or not show the text indicating the p value
`show.CI`	a logical vector indicating whether or not show the text indicating the confidence interval
`sig.level`	a numeric value of upper limit of p value of showing variables
`cex`	A numerical value giving the amount by which plotting OR/HR symbols should be magnified relative to the default, defaulting 1.2.
`lwd`	The line width, a positive number, defaulting to 2.
`pch`	Either an integer specifying a symbol or a single character to be used as the default in plotting OR/HR points.
`col`	A specification for the default plotting color.
`...`	arguments to be passed to plot

Value

This function return NULL invisibly and draw graphs

Examples

```
require(survival)
attach(colon)
colon$TS=Surv(time,status==1)
out=mycph(TS~.,data=colon)
out
HRplot(out)
## Not run:
HRplot(out,type=1,pch=2,col=c("blue","red"))
HRplot(out,type=2,show.CI=TRUE,pch=2,cex=2,main="Hazard ratios of all individual variables")

## End(Not run)
```

`my.chisq.test` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
my.chisq.test(x, y, mydata, catMethod = 2)
```

Arguments

`x`	a vector
`y`	a vector
`mydata`	A data.frame
`catMethod`	An integer indicating methods for categorical variables. Possible values in methods are **0** Perform chisq.test first. If warning present, perform fisher test **1** Perform chisq.test without continuity correction **2** Perform chisq.test with continuity correction **3** perform fisher.test **4** perform prop.trend test Default value is 2.

`my.t.test` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
my.t.test(y, x)
```

Arguments

`y`	a vector
`x`	a numeric vector

`mycph` *Perform coxph of individual expecting variables*

Description

Perform coxph of individual expecting variables

Usage

```
mycph(formula, data, digits = 2)
```

Arguments

`formula`	An object of class "formula". Left side of ~ must be a variable of class Surv and the right side of ~ must have variables in an additive way.
`data`	A data.frame contains data for analysis.
`digits`	An integer indicating the number of decimal places (round) or significant digits to be used. Default value is 2.

Value

a data.frame consist of hazard ratio and 95% confidence intervals and the p values.

Examples

```
require(survival)
data(cancer)
attach(colon)
colon$TS=Surv(time,status==1)
out=mycph(TS~.,data=colon)
out
HRplot(out,type=2,show.CI=TRUE,main="Hazard ratios of all individual variables")
```

`mycsv` *Export to csv file for class "mytable" or "cbind.mytable"*

Description

Export to csv file for class "mytable" or "cbind.mytable"

Usage

```
mycsv(x, row.names = FALSE, ...)
```

Arguments

`x`	An object of class "mytable" or "cbind.mytable"
`row.names`	either a logical value indicating whether the row names of x are to be written along with x, or a character vector of row names to be written.
`...`	further arguments passed to or from other methods.

Examples

```
## Not run:
require(moonBook)
res=mytable(sex~age+DM,data=acs)
mycsv(res,file="test.csv")
mycsv(summary(res),file="testsummary.csv")

## End(Not run)
```

`mycsv.cbind.mytable` *Export to csv file for class "cbind.mytable"*

Description

Export to csv file for class "cbind.mytable"

Usage

```
## S3 method for class 'cbind.mytable'
mycsv(x, row.names = FALSE, ...)
```

Arguments

`x`	An object of class "cbind.mytable" a result of a call to `mytable`
`row.names`	either a logical value indicating whether the row names of x are to be written along with x, or a character vector of row names to be written.
`...`	further arguments passed to or from other methods.

Examples

```
## Not run:
require(moonBook)
res1=mytable(sex+Dx~age+DM,data=acs)
mycsv(res1,file="test1.csv")
mycsv(summary(res1),file="testsummary1.csv")

## End(Not run)
```

`mycsv.mytable` *Export to csv file for class "mytable"*

Description

Export to csv file for class "mytable"

Usage

```
## S3 method for class 'mytable'
mycsv(x, row.names = FALSE, ...)
```

Arguments

`x`	An object of class "mytable" a result of a call to `mytable`
`row.names`	either a logical value indicating whether the row names of x are to be written along with x, or a character vector of row names to be written.
`...`	further arguments passed to or from other methods.

Examples

```
## Not run:
require(moonBook)
res=mytable(sex~age+DM,data=acs)
mycsv(res,file="test.csv")
mycsv(summary(res),file="testsummary.csv")
mycsv=function(x,row.names=FALSE) UseMethod("mycsv")

## End(Not run)
```

`myhtml` *Export to html file for class "mytable" or "cbind.mytable" of "data.frame"*

Description

Export to html file for class "mytable" or "cbind.mytable" of "data.frame"

Usage

```
myhtml(x, caption = NULL, rownames = TRUE)

## Default S3 method:
myhtml(x, caption = NULL, rownames = TRUE)

## S3 method for class 'mytable'
```

```
myhtml(x, caption = NULL, rownames = TRUE)

## S3 method for class 'cbind.mytable'
myhtml(x, caption = NULL, rownames = TRUE)
```

Arguments

`x`	An object of class "mytable" or "cbind.mytable"
`caption`	A character
`rownames`	A logical value whether or not include rownames in table

Methods (by class)

- `default`:
- `mytable`:
- `cbind.mytable`:

Examples

```
require(moonBook)
res=mytable(sex~age+Dx,data=acs)
myhtml(res)
res1=mytable(sex+Dx~.,data=acs)
myhtml(res1)
x=head(iris)
myhtml(x)
myhtml(x,caption="Table 1. myhtml Test")
myhtml(x,caption="Table 1. myhtml Test",rownames=FALSE)
```

`myhtmlHead` *Print my html style*

Description

Print my html style

Usage

```
myhtmlHead()
```

`mylatex` *Exporting "cbind.mytable","mytable" to LaTeX format*

Description

Exporting "cbind.mytable","mytable" to LaTeX format

Usage

```
mylatex(
  myobj,
  size = 5,
  caption = NULL,
  caption.placement = "top",
  caption.position = "c"
)

## Default S3 method:
mylatex(
  myobj,
  size = 5,
  caption = NULL,
  caption.placement = "top",
  caption.position = "c"
)

## S3 method for class 'mytable'
mylatex(
  myobj,
  size = 5,
  caption = NULL,
  caption.placement = "top",
  caption.position = "c"
)

## S3 method for class 'cbind.mytable'
mylatex(
  myobj,
  size = 5,
  caption = NULL,
  caption.placement = "top",
  caption.position = "c"
)
```

Arguments

`myobj`	An object of class 'mytable'

`size`	An integer indicating font size, defaulting is 5.
`caption`	A character
`caption.placement`	The caption will be have placed at the top of the table if caption.placement is "top" and at the bottom of the table if it equals "bottom". Default value is "top".
`caption.position`	The caption will be have placed at the center of the table if caption.position is "center" or "c", and at the left side of the table if it equals "left" or "l", and at the right side of the table if it equals "right" or "r". Default value is "center".

Methods (by class)

- `default`: Exporting "cbind.mytable","mytable" to LaTeX format
- `mytable`: Exporting "cbind.mytable","mytable" to LaTeX format
- `cbind.mytable`: Exporting "cbind.mytable","mytable" to LaTeX format

Examples

```
require(moonBook)
out=mytable(sex~.,data=acs)
mylatex(out)
out1=mytable(sex+Dx~.,data=acs)
mylatex(out1,size=6)
```

`mytable` *Produce table for descriptive statistics*

Description

Produce table for descriptive statistics by groups for several variables easily. Depending on the nature of these variables, different descriptive statistical methods were used(t-test, ANOVA,Kruskal-Wallis, chisq, Fisher,...)

Usage

```
mytable(x, ...)

## S3 method for class 'formula'
mytable(x, ...)

## S3 method for class 'data.frame'
mytable(x, ...)
```

Arguments

`x`	An R object, formula or data.frame
`...`	arguments to be passed to `mytable_sub`

Methods (by class)

- `formula`: S3 method for formula
- `data.frame`: S3 method for data.frame

Examples

```
mytable(acs)
mytable(~age+sex,data=acs)
mytable(Dx~age+sex+height+weight+TC+TG+HDLC,data=acs,method=3,digits=2)
mytable(am+cyl~.,data=mtcars)
out=mytable(sex~.,data=acs)
out
summary(out)
## Not run:
require(ztable)
ztable(out)

## End(Not run)
mytable(acs)
```

`mytable2` *Produce combined table for descriptive statistics*

Description

Produce table for descriptive statistics by two grouping variables for several variables easily. Depending on the nature of these variables, different descriptive statistical methods were used(t-test, ANOVA,Kruskal-Wallis, chisq, Fisher,...)

Usage

```
mytable2(
  formula,
  data,
  use.labels = TRUE,
  use.column.label = TRUE,
  max.ylev = 5,
  maxCatLevel = 20,
  digits = 2,
  method = 1,
  catMethod = 2,
  show.all = FALSE,
  exact = FALSE,
  show.total = FALSE,
  origData = NULL
)
```

Arguments

`formula`	An object of class "formula". Left side of ~ must contain two grouping variables in an additive way(e.g. sex+group~), and the right side of ~ must have variables in an additive way.
`data`	A data.frame contains data for analysis
`use.labels`	Logical. Whether or not use labels.
`use.column.label`	Logical. Whether or not use column labels.
`max.ylev`	An integer indicating the maximum number of levels of grouping variable ('y'). If a column have unique values less than max.ylev it is treated as a categorical variable. Default value is 5.
`maxCatLevel`	An integer indicating the maximum number of unique levels of categorical variable. If a column have unique values more than maxCatLevel, categorical summarization will not be performed.
`digits`	An integer indicating the number of decimal places (round) or significant digits to be used. Default value is 1.
`method`	An integer indicating methods for continuous variables. Possible values in methods are **1** forces analysis as normal-distributed **2** forces analysis as continuous non-normal **3** performs a Shapiro-Wilk test to decide between normal or non-normal Default value is 1.
`catMethod`	An integer indicating methods for categorical variables. Possible values in methods are **0** Perform chisq.test first. If warning present, perform fisher test **1** Perform chisq.test without continuity correction **2** Perform chisq.test with continuity correction **3** perform fisher.test **4** perform prop.trend test Default value is 2.
`show.all`	A logical value indicating whether or not all statistical values have to be shown in table. Default value is FALSE.
`exact`	A logical value indicating whether or not permit call with approximate parameter. If true, only exact column name permitted.Default value is FALSE.
`show.total`	A logical value indicating whether or not show total group value. Default value is FALSE.
`origData`	A data.frame contains data for analysis

Value

An object of class "cbind.mytable"

`mytable2df` *Convert mytable object to data.frame*

Description

Add N number into data.frame

Usage

```
mytable2df(x)
```

Arguments

`x`	An object of class "mytable" a result of a call to `mytable`

Value

a data.frame with N number

`mytable2html` *Prepare mytable object to data.frame ready to html*

Description

Add N number into data.frame

Usage

```
mytable2html(x)
```

Arguments

`x`	An object of class "mytable" a result of a call to `mytable`

Value

a data.frame with N number

`mytable_df` *make mytable from data.frame*

Description

make mytable from data.frame

Usage

```
mytable_df(
  x,
  use.labels = TRUE,
  use.column.label = TRUE,
  max.ylev = 5,
  maxCatLevel = 20,
  digits = 1,
  method = 1,
  show.all = FALSE
)
```

Arguments

`x`	A data.frame
`use.labels`	Logical. Whether or not use labels.
`use.column.label`	Logical. Whether or not use column labels.
`max.ylev`	An integer indicating the maximum number of levels of grouping variable. If a column have unique values less than max.ylev it is treated as a categorical variable. Default value is 5.
`maxCatLevel`	An integer indicating the maximum number of unique levels of categorical variable. If a column have unique values more than maxCatLevel, categorical summarization will not be performed.
`digits`	An integer indicating the number of decimal places (round) or significant digits to be used. Default value is 1.
`method`	An integer indicating methods for continuous variables. Possible values in methods are **1** forces analysis as normal-distributed **2** forces analysis as continuous non-normal **3** performs a Shapiro-Wilk test to decide between normal or non-normal Default value is 1.
`show.all`	A logical value indicating whether or not all statistical values have to be shown in table. Default value is FALSE.

Value

An object of class "mytable.df". 'print' returns a table for descriptive statistics.

`mytable_sub` *Produce table for descriptive statistics*

Description

Produce table for descriptive statistics by groups for several variables easily. Depending on the nature of these variables, different descriptive statistical methods were used(t-test, ANOVA,Kruskal-Wallis, chisq, Fisher,...)

Usage

```
mytable_sub(
  x,
  data,
  use.labels = TRUE,
  use.column.label = TRUE,
  max.ylev = 5,
  maxCatLevel = 20,
  digits = 1,
  method = 1,
  catMethod = 2,
  show.all = FALSE,
  exact = FALSE,
  show.total = FALSE,
  missing = FALSE
)
```

Arguments

`x`	An object of class "formula". Left side of ~ must contain the name of one grouping variable or two grouping variables in an additive way(e.g. sex+group~), and the right side of ~ must have variables in an additive way.
`data`	A data.frame contains data for analysis
`use.labels`	Logical. Whether or not use labels.
`use.column.label`	Logical. Whether or not use column labels.
`max.ylev`	An integer indicating the maximum number of levels of grouping variable ('y'). If a column have unique values less than max.ylev it is treated as a categorical variable. Default value is 5.
`maxCatLevel`	An integer indicating the maximum number of unique levels of categorical variable. If a column have unique values more than maxCatLevel, categorical summarization will not be performed.
`digits`	An integer indicating the number of decimal places (round) or significant digits to be used. Default value is 1.

`method`	An integer indicating methods for continuous variables. Possible values in methods are **1** forces analysis as normal-distributed **2** forces analysis as continuous non-normal **3** performs a Shapiro-Wilk test to decide between normal or non-normal Default value is 1.
`catMethod`	An integer indicating methods for categorical variables. Possible values in methods are **0** Perform chisq.test first. If warning present, perform fisher test **1** Perform chisq.test without continuity correction **2** Perform chisq.test with continuity correction **3** perform fisher.test **4** perform prop.trend test Default value is 2.
`show.all`	A logical value indicating whether or not all statistical values have to be shown in table. Default value is FALSE.
`exact`	A logical value indicating whether or not permit call with approximate parameter. If true, only exact column name permitted.Default value is FALSE.
`show.total`	A logical value indicating whether or not show total group value. Default value is FALSE.
`missing`	A logical value indicating whether or not perform missing data analysis. Default value is FALSE.

Value

An object of class "mytable". 'print' returns a table for descriptive statistics. 'summary' returns a table with all statistical values.

`mytable_sub2` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
mytable_sub2(
  y,
  x,
  data,
  max.ylev = 5,
  maxCatLevel = 20,
```

```
    method = 1,
    catMethod = 2,
    show.total = FALSE,
    origData
  )
```

Arguments

`y`	a vector
`x`	a vector
`data`	a data.frame
`max.ylev`	an integer
`maxCatLevel`	an integer
`method`	an integer
`catMethod`	an integer
`show.total`	a logical value
`origData`	a data.frame

`num_summary` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
num_summary(x)
```

Arguments

`x`	a numeric vector

`obj2linecount` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
obj2linecount(myobj)
```

Arguments

`myobj`	an R object

`ORplot` *Plot for odds ratios for a S3 object of glm*

Description

Plot for odds ratios for a S3 object of glm

Usage

```
ORplot(
  x,
  type = 1,
  xlab = "",
  ylab = "",
  show.OR = TRUE,
  show.CI = FALSE,
  sig.level = 1,
  cex = 1.2,
  lwd = 2,
  pch = 18,
  col = NULL,
  ...
)
```

Arguments

`x`	A S3 object of glm
`type`	an integer defining the shape of plots; default value is 1
`xlab`	label for the horizontal-axis; defaults to "Odds Ratios"
`ylab`	label for the vertical axis; defaults to "".
`show.OR`	A logical value; Whether or not show p values on plot
`show.CI`	A logical value; Whether or not show 95% CI values on plot
`sig.level`	A numeric value of upper limit of p value of showing variables
`cex`	A numerical value giving the amount by which plotting OR/HR symbols should be magnified relative to the default, defaulting 1.2.
`lwd`	The line width, a positive number, defaulting to 2.
`pch`	Either an integer specifying a symbol or a single character to be used as the default in plotting OR/HR points.
`col`	A specification for the default plotting color.
`...`	arguments to be passed to plot

Value

This function return NULL invisibly and draw graphs

Examples

```
require(survival)
data(colon)
out1=glm(status~sex+age+rx+obstruct+node4,data=colon)
out2=glm(status~rx+node4,data=colon)
ORplot(out1,type=2,show.CI=TRUE,xlab="This is xlab",main="Main Title")
ORplot(out2,type=1,main="Main Title")
ORplot(out1,type=2,show.CI=TRUE,main="Main Title")
ORplot(out1,type=3,show.CI=TRUE,main="Main Title",sig.level=0.05)
ORplot(out1,type=3,show.CI=TRUE,main="Main Title",sig.level=0.05,
      pch=1,cex=2,lwd=4,col=c("red","blue"))
```

`ORplot.sub` *A sub function for ORplot and HRplot*

Description

Plot for odds ratios for a S3 object of glm

Usage

```
ORplot.sub(
  result,
  type = 1,
  xlab = "",
  ylab = "",
  show.OR = TRUE,
  show.CI = FALSE,
  sig.level = 1,
  cex = 1.2,
  lwd = 2,
  pch = 18,
  col = NULL,
  ...
)
```

Arguments

`result`	A resultant data.frame of function extractOR
`type`	an integer defining the shape of plots; default value is 1
`xlab`	label for the horizontal-axis; defaults to "Odds Ratios"
`ylab`	label for the vertical axis; defaults to "".
`show.OR`	A logical value; Whether or not show p values on plot
`show.CI`	A logical value; Whether or not show 95% CI values on plot
`sig.level`	A numeric value of upper limit of p value of showing variables

`cex`	A numerical value giving the amount by which plotting OR/HR symbols should be magnified relative to the default, defaulting 1.2.
`lwd`	The line width, a positive number, defaulting to 2.
`pch`	Either an integer specifying a symbol or a single character to be used as the default in plotting OR/HR points.
`col`	A specification for the default plotting color.
`...`	Further arguments to be passed to plot

Value

This function return NULL invisibly and draw graphs

`p2sig` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
p2sig(value)
```

Arguments

`value`	a numeric vector

`print.cbind.mytable` *Print function for class "cbind.mytable"*

Description

Print function for class "cbind.mytable"

Usage

```
## S3 method for class 'cbind.mytable'
print(x, ...)
```

Arguments

`x`	an object of class "cbind.mytable", a result of a call to `cbind.mytable`
`...`	further arguments passed to or from other methods.

`print.mytable` *Print function for class "mytable"*

Description

Print function for class "mytable"

Usage

```
## S3 method for class 'mytable'
print(x, ...)
```

Arguments

`x`	An object of class "mytable", a result of a call to `mytable`
`...`	further arguments passed to or from other methods.

`print.mytable.df` *Print an object of mytable.df*

Description

Print an object of mytable.df

Usage

```
## S3 method for class 'mytable.df'
print(x, ...)
```

Arguments

`x`	An object of class mytable.df
`...`	Further arguments

`printmytable2` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
printmytable2(obj, digits = 1)
```

Arguments

`obj`	an object
`digits`	an integer

`r` *Subfunction used in mylatex*

Description

Subfunction used in mylatex

Usage

```
r(string)
```

Arguments

`string`	a character vector

`radial` *Demographic data of 115 patients performing IVUS(intravascular ultrasound) examination of a radial artery.*

Description

A dataset containing demographic data and laboratory data of 115 patients performing IVUS(intravascular ultrasound) examination of a radial artery after tansradial coronary angiography.

Format

A data frame with 115 rows and 15 variables:

male if Male, 1; if Female 0

age patient age in years

height height in centimeter

weight weight in kilogram

HBP history of hypertension, 1 for yes or 0 for no

DM history of diabetes mellitus, 1 for yes or 0 for no

smoking history of smoking, One of the followings : "non-smoker","ex-smoker","smoker"

TC total cholesterol level in mg/dL

TG triglyceride level in mg/dL

HDL high density lipoprotein cholesterol level in mg/dL

LDL low density lipoprotein cholesterol level in mg/dL

hsCRP high-sensitive C reactive protein

NTAV normalized total atheroma volume measured by IVUS in cubic mm

PAV percent atheroma volume in percentage

sex Factor with two levels; "Male" or "Female"

`rank2group` *rank a numeric vector and returns a new ordinal vector*

Description

rank a numeric vector and returns a new ordinal vector

Usage

```
rank2group(y, k = 4)
```

Arguments

`y`	a numeric vector
`k`	a integer specifies how many groups you want to classify. default value is 4

Value

a ordinal vector(numeric) with the same length of y

Examples

```
require(ggplot2)
data(diamonds)
diamonds$PriceGroup=rank2group(diamonds$price,4)
table(diamonds$PriceGroup)
aggregate(price~PriceGroup,data=diamonds,range)

diamonds$PriceGroup3=rank2group(diamonds$price,3)
table(diamonds$PriceGroup3)
aggregate(price~PriceGroup3,data=diamonds,range)
diamonds$PriceGroup5=rank2group(diamonds$price,5)
table(diamonds$PriceGroup5)
aggregate(price~PriceGroup5,data=diamonds,range)
```

`reprint` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
reprint(x, times)
```

Arguments

`x`	a character vector
`times`	an integer

`space` *Internal mytable functions*

Description

Internal mytable functions These are not to be called by the user

Usage

```
space(num)
```

Arguments

`num`	an integer

`summary.cbind.mytable` *Summarizing function for class "cbind.mytable"*

Description

Summarizing function for class "cbind.mytable"

Usage

```
## S3 method for class 'cbind.mytable'
summary(object, ...)
```

Arguments

`object`	An object of class "cbind.mytable", a result of a call `mytable`
`...`	further arguments passed to or from other methods.

Examples

```
out=mytable(am+cyl~.,data=mtcars)
summary(out)
```

`summary.mytable` *Summarizing function for class "mytable"*

Description

Summarizing function for class "mytable"

Usage

```
## S3 method for class 'mytable'
summary(object, ...)
```

Arguments

`object`	An object of class "mytable", a result of a call `mytable`
`...`	further arguments passed to or from other methods.

Examples

```
out=mytable(am~.,data=mtcars)
summary(out)
```

`validColname` *Find valid string among character vector from approximate string*

Description

Find valid string among character vector from approximate string

Usage

```
validColname(pattern, x)
```

Arguments

`pattern`	character string to be matched in the given character
`x`	a character vector where matches are sought

Value

returns NA in case of no matched string found or a character string in string vector x

Examples

```
a="dx"
b=c("Age","Sex","Dx")
validColname(a,b)
```

Index