

Package ‘quickReg’

August 27, 2016

Title Build Regression Models Quickly and Display the Results Using 'ggplot2'

Version 1.0.0

Description A set of functions to extract results from regression models and plot the effect size using 'ggplot2' seamlessly. While 'broom' is useful to convert statistical analysis objects into tidy data frames, 'coefplot' is adept at showing multivariate regression results. With specific outcome, this package could build regression models automatically, extract results into a data frame and provide a quicker way to summarize models' statistical findings using 'ggplot2'.

Depends R (>= 3.0.0)

Imports ggplot2 (>= 2.0.0), survival, psych, utils, stats, nortest

Suggests ggthemes, knitr, rmarkdown

License GPL-2

BugReports Xikun Han <hanxikun2014@163.com>

LazyData true

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Xikun Han [aut, cre]

Maintainer Xikun Han <hanxikun2014@163.com>

Repository CRAN

Date/Publication 2016-08-27 15:51:45

R topics documented:

dataframe	2
dataframe.reg	3
detail	3
detail.reg	4
diabetes	5
display	5

display.data.frame	6
display.reg	7
plot.reg	7
reg	8

Index	10
--------------	-----------

dataframe	<i>Generic function for class 'reg'</i>
-----------	---

Description

Generic function for class 'reg' to extracte concentrated results of regression models.

Usage

```
dataframe(x, ...)
```

Arguments

x	A reg object
...	additional arguments

Value

A data frame of univariate regression result

See Also

[dataframe.reg](#), [reg](#)

Examples

```
data(diabetes)
head(diabetes)

reg_coxph<-reg(data = diabetes, x = c(1,4, 6),
y = 5, time = 2, factor = c(1, 3, 4), model = 'coxph')

dataframe(reg_coxph,save=FALSE)
```

dataframe.reg	<i>Retrieve a data.frame result of univariate regression</i>
---------------	--

Description

Return a concentrated result of univariate regression models.

Usage

```
## S3 method for class 'reg'  
dataframe(x, save = FALSE, file = NULL, sep = ",",  
          row.names = FALSE, ...)
```

Arguments

x	A reg object
save	A logical, whether to save the concentrated result
file	A character string naming a file, see write.table for more details, the default filepath is current working directory and the filename is the current time.
sep, row.names, ...	See write.table for more details

See Also

[reg](#)

Examples

```
reg_glm<-reg(data = diabetes, x=c(1:4),y = 5,  
            factor = c(1, 3, 4), model = 'glm')  
  
dataframe(reg_glm)  
# dataframe(reg_glm, save = TRUE)  
# dataframe(reg_glm, file = "C:/reg_glm_out.csv")
```

detail	<i>Generic function for class 'reg'</i>
--------	---

Description

Generic function for class 'reg' to extract every regression model result.

Usage

```
detail(x)
```

Arguments

x A reg object

See Also

[detail.reg](#), [reg](#)

Examples

```
data(diabetes)
head(diabetes)
reg_coxph<-reg(data = diabetes, x = c(1,4, 6), y = 5, time = 2, factor = c(1, 4), model = 'coxph')
detail(reg_coxph)
```

detail.reg

Retrieve detail results of univariate regression models

Description

Retrieve detail results of univariate regression models

Usage

```
## S3 method for class 'reg'
detail(x)
```

Arguments

x A reg object

See Also

[reg](#)

Examples

```
data(diabetes)
head(diabetes)

reg_glm<-reg(data = diabetes, y = 5, factor = c(1, 3, 4), model = 'glm')
detail(reg_glm)
```

`diabetes`*A hypothetical dataset*

Description

A hypothetical dataset extracted from package ‘PredictABEL’

Usage

```
diabetes
```

Format

A data frame with 1000 rows and 14 variables:

- sex: 1=male, 2=female
- age: age of the participants(years)
- smoking: 0=non smoker, 1=smoker
- education: 0=without bachelor degree, 1=bachelor degree or above
- diabetes: diabetes mellitus, 0= health, 1= diabetes
- BMI: body mass index (kg/cm2)
- systolic: systolic blood pressure(mmHg)
- diastolic: diastolic blood pressure(mmHg)
- ...: other genetic informatin, see the [ExampleData](#) in [PredictABEL-package](#) .

See Also

[ExampleData](#)

`display`*Generic function to display a data.frame or ‘reg’ object*

Description

`display` is a generic function to display summary information of variables in a data.frame or summary result of ‘reg’ class

Usage

```
display(x, ...)
```

Arguments

x an object to display, a data.frame or a 'reg' class
 ... additional arguments

See Also

[display.data.frame](#), [display.reg](#)

display.data.frame *Display summary information of variables in a data.frame*

Description

Display count, frequency or mean, standard deviation and test of normality, etc.

Usage

```
## S3 method for class 'data.frame'
display(x = NULL, col = NULL, normtest = NULL,
        useNA = "ifany", discrete_limit = 10, exclude_discrete = TRUE, ...)
```

Arguments

x A data.frame
 col Column indices or names of the variables in the dataset to display, the default columns are all the variables
 normtest A character indicating test of normality, the default method is [shapiro.test](#) when sample size no more than 5000, otherwise [lillie.test](#) Kolmogorov-Smirnov is used, see package **nortest** for more methods. Use 'shapiro.test', 'lillie.test', 'ad.test', etc to specify methods.
 useNA Whether to include NA values in the table, see [table](#) for more details
 discrete_limit A numeric defining the minimal of unique value to display the variable as count and frequency
 exclude_discrete Logical, whether to exclude discrete variables with more unique values specified by discrete_limit
 ... additional arguments

See Also

[display.reg](#), [display](#)

Examples

```
data(diabetes)
head(diabetes)
display(diabetes, 1:2)
display(diabetes, 1:11, normtest = "lillie.test")
```

display.reg	<i>summary the result of univariate regression</i>
-------------	--

Description

return summary information of univariate regression.

Usage

```
## S3 method for class 'reg'  
display(x, cuts = c(0.001, 0.01, 0.05, 0.1, 1),  
       all_var = TRUE, ...)
```

Arguments

x	A reg object
cuts	A vector of significance values.
all_var	A logical, whether to display all variables at specific p value cut-off, if FALSE, only display the first 10 variables
...	additional arguments

Value

Summary information of significant variables.

See Also

[dataframe.reg](#), [display](#), [display.reg](#)

Examples

```
reg_glm<-reg(data = diabetes, y = 5, factor = c(1, 3, 4), model = 'glm')  
display(reg_glm)  
display(reg_glm, all_var = FALSE)
```

plot.reg	<i>plot the result of univariate regression result</i>
----------	--

Description

plot coefficients, OR or HR of univariate regressions.

Usage

```
## S3 method for class 'reg'  
plot(x, limits = c(NA, NA), sort = "order", ...)
```

Arguments

x	A reg object
limits	A numeric vector of length two providing limits of the scale. Use NA to refer to the existing minimum or maximum value.
sort	A character determining the order of variables to plot, 'alphabetical' or 'order'. The later is the default to sort variables according to their effect size.
...	additional arguments

See Also

[reg](#), [dataframe.reg](#), [detail.reg](#)

Examples

```
reg_glm<-reg(data = diabetes, y = 5, factor = c(1, 3, 4), model = 'glm')
plot(reg_glm)
plot(reg_glm, limits = c(NA, 3))
```

 reg

Apply univariate regression models

Description

Apply general linear model, generalized linear model, cox regression model, etc.

Usage

```
reg(data = NULL, x = NULL, y = NULL, factor = NULL, model = NULL,
     time = NULL, ...)
```

Arguments

data	A data.frame
x	Column indices or names of the variables to be included in univariate analysis, the default columns are all the variables except dependent
y	Column indice of dependent variable or name
factor	Column indices of the variables or names to be treated as factor
model	Univariate analysis method, see lm , glm , coxph
time	Column indice of survival time or name, used in cox regression, see coxph for more details
...	Further arguments passed to regression model

Value

The return result is a list including two components, the first part is a detailed analysis result, the second part is a concentrated result in a data.frame

Examples

```
reg_glm<-reg(data = diabetes, x = c(1:4, 6), y = 5, factor = c(1, 3, 4), model = 'glm')
## subset result like a list
reg_glm$detail
reg_glm$dataframe
reg_glm[2]
reg_glm$detail[2:4]
## other methods
reg(data = diabetes, x = c(1, 3:6), y = 10, factor = c(1, 3, 4), model = 'lm')
reg(data = diabetes, x = c( "sex","education","BMI"), y = "diabetes",
time ="age", factor = c("sex","smoking","education"), model = 'coxph')
```

Index

*Topic **datasets**

diabetes, 5

coxph, 8

dataframe, 2

dataframe.reg, 2, 3, 7, 8

detail, 3

detail.reg, 4, 4, 8

diabetes, 5

display, 5, 6, 7

display.data.frame, 6, 6

display.reg, 6, 7, 7

ExampleData, 5

glm, 8

lillie.test, 6

lm, 8

plot.reg, 7

reg, 2-4, 8, 8

shapiro.test, 6

table, 6

write.table, 3