Package: quickReg (via r-universe)

September 16, 2024

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

Version 1.5.0

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), rlang (>= 0.1.2), survival, psych, utils, stats, nortest, dplyr

Suggests ggthemes, knitr, rmarkdown, testthat

License GPL-2

LazyData true

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Xikun Han [aut, cre]

Maintainer Xikun Han `<hanxikun2014@163.com>`

Date/Publication 2017-09-28 06:41:12 UTC

Repository https://han-xikun.r-universe.dev

RemoteUrl https://github.com/cran/quickReg

RemoteRef HEAD

RemoteSha 67d634349547c2f798305b1104b31b5183dc73e3

Contents

`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 information, see the `ExampleData` in `PredictABEL-package` .

See Also

`ExampleData`

`display_table` *Display a table used in paper*

Description

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

Usage

```
display_table(data = NULL, variables = NULL, group = NULL,
  mean_or_median = "mean", addNA = TRUE, table_margin = 2,
  discrete_limit = 10, exclude_discrete = TRUE, save_to_file = NULL,
  normtest = NULL, fill_variable = FALSE)

display_table_group(data = NULL, variables = NULL, group = NULL,
  super_group = NULL, group_combine = FALSE, mean_or_median = "mean",
  addNA = TRUE, table_margin = 2, discrete_limit = 10,
  exclude_discrete = TRUE, normtest = NULL, fill_variable = FALSE)
```

Arguments

`data`	A data.frame
`variables`	Column indices or names of the variables in the dataset to display, the default columns are all the variables except group variable
`group`	Column indices or names of the first subgroup variables. Must provide.
`mean_or_median`	A character to specify mean or median to used for continuous variables, either "mean" or "median". The default is "mean"
`addNA`	Whether to include NA values in the table, see `table` for more details
`table_margin`	Index of generate margin for, see `prop.table` for more details
`discrete_limit`	Defining the minimal of unique value to display the variable as count and frequency, the default is 10
`exclude_discrete`	Logical, whether to exclude discrete variables with more unique values specified by discrete_limit
`save_to_file`	A character, containing file name or path
`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.
`fill_variable`	A logical, whether to fill the variable column in result, the default is FALSE
`super_group`	Column indices or names of the further subgroup variables.
`group_combine`	A logical, subgroup analysis for combination of variables or for each variable. The default is FALSE (subgroup analysis for each variable)

Functions

- `display_table_group`: Allow more subgroup analysis, see the package vignette for more details

Note

The return table is a data.frame.

- P.value1 is ANOVA P value for continuous variables and chi-square test P value for discrete variables

- P.value2 is Kruskal-Wallis test P value for continuous variables and fisher test P value for discrete variables if expected counts less than 5

- normality is normality test P value for each group

Examples

```
## Not run:
data(diabetes)
head(diabetes)
library(dplyr);library(rlang)
result_1<-diabetes %>%
 group_by(sex) %>%
 do(display_table(data=.,variables=c("age","smoking"),group="CFHrs2230199")) %>%
 ungroup()
result_2<-display_table_group(data=diabetes,variables=c("age","smoking"),
group="CFHrs2230199",super_group = "sex")
identical(result_1,result_2)
result_3<-display_table_group(data=diabetes,variables=c("age","education"),
group=c("smoking"),super_group = c("CFHrs2230199","sex"))
result_4<-display_table_group(data=diabetes,variables=c("age","education"),
group=c("smoking"),super_group = c("CFHrs2230199","sex"),group_combine=TRUE)

## End(Not run)
```

`plot.reg` *plot the result of regression result*

Description

plot coefficients, OR or HR of regression models.

Usage

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

plot_reg(x, limits = c(NA, NA), sort = "order", title = NULL,
  remove = TRUE, term = NULL, center = NULL, low = NULL, high = NULL,
  model = NULL, ...)
```

Arguments

`x`	A reg, reg_y or reg_x object without covariates information, 'cov_show=FALSE'
`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.
`title`	title of plot
`remove`	A logical, whether to remove infinite and NA value. The default is TRUE
`...`	additional arguments. When using your own regression results rather than from 'quickReg', please provide 'term','center','lower', 'high' and 'model' for plot.
`term`	A character of x axis variable in plot
`center`	A character of coefficient, OR or HR variable in plot
`low`	A character of lower confidence interval variable
`high`	A character of upper confidence interval variable
`model`	A character of model, "lm", "glm" or "coxph"

See Also

`reg`, `reg_x`, `reg_y`

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` *Build regression models*

Description

Build general linear model, logistic regression model, cox regression model with one or more dependent variables. Allow regression based on subgroup variables.

Usage

```
reg(data = NULL, x = NULL, y = NULL, group = NULL, cov = NULL,
  factors = NULL, model = NULL, time = NULL, cov_show = FALSE,
  confint_glm = "default", group_combine = FALSE)
```

Arguments

`data`	A data.frame to build the regression model.
`x`	Integer column indices or names of the variables to be included in univariate analysis. If `NULL`, the default columns are all the variables except ‘y‘, ‘group‘, ‘time‘ and ‘cov‘.
`y`	Integer column indice or name of dependent variables, integer or character, allow more than one dependent variables
`group`	Integer column indice or name of subgroup variables.
`cov`	Integer column indices or name of covariate variables
`factors`	Integer column indices or names of variables to be treated as factor
`model`	regression model, see `lm`, `glm`, `coxph` for more details
`time`	Integer column indices or name of survival time, used in cox regression, see `coxph` for more details
`cov_show`	A logical, whether to create covariates result, default FALSE
`confint_glm`	A character, ’default’ or ’profile’. The default method for ’glm’ class to compute confidence intervals assumes asymptotic normality `confint`, you can also use profile likelihood method `confint.glm`, but it is pretty slow. In this case you could specify ’default’ for speed.
`group_combine`	A logical, subgroup analysis for combination of group variables or each group variables. The default is FALSE (subgroup analysis for each group variable)

Value

The return result is a concentrated result in a data.frame.

`reg_x` *Build regression models only one dependent variable*

Description

Build general linear model, generalized linear model, cox regression model with only one dependent variables.

Usage

```
reg_x(data = NULL, x = NULL, y = NULL, cov = NULL, factors = NULL,
  model = NULL, time = NULL, cov_show = FALSE, detail_show = FALSE,
  confint_glm = "default", save_to_file = NULL)
```

Arguments

`data`	A data.frame to build the regression model.
`x`	Integer column indices or names of the variables to be included in univariate analysis. If `NULL`, the default columns are all the variables except ‘y‘, ‘time‘ and ‘cov‘.
`y`	Integer column indice or name of dependent variable, only one integer or character
`cov`	Integer column indices or name of covariate variables
`factors`	Integer column indices or names of variables to be treated as factor
`model`	regression model, see `lm`, `glm`, `coxph` for more details
`time`	Integer column indices or name of survival time, used in cox regression, see `coxph` for more details
`cov_show`	A logical, whether to create covariates result, default FALSE
`detail_show`	A logical, whether to create each regression result, default FALSE. If TRUE, with many regressions, the return result could be very large.
`confint_glm`	A character, ’default’ or ’profile’. The default method for ’glm’ class to compute confidence intervals assumes asymptotic normality `confint`, you can also use profile likelihood method `confint.glm`, but it is pretty slow. In this case you could specify ’default’ for speed.
`save_to_file`	A character, containing file name or path

Value

If detail_show is TRUE, 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. Otherwise, only return concentrated result in a data.frame.

Examples

```
reg_glm<-reg_x(data = diabetes, x = c(1:4, 6), y = 5, factors = c(1, 3, 4), model = 'glm')
##  other methods
fit<-reg_x(data = diabetes, x = c(1, 3:6), y = "age", factors = c(1, 3, 4), model = 'lm')
fit<-reg_x(data = diabetes, x = c( "sex","education","BMI"), y = "diabetes",
time ="age", factors = c("sex","smoking","education"), model = 'coxph')
```

`reg_y` *Build regression models with more than one dependent variable*

Description

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

Usage

```
reg_y(data = NULL, x = NULL, y = NULL, cov = NULL, factors = NULL,
  model = NULL, time = NULL, confint_glm = "default", cov_show = FALSE)
```

Arguments

`data`	A data.frame
`x`	Integer column indices or names of the variables to be included in univariate analysis, the default columns are all the variables except ‘y‘ and ‘time‘ and ‘cov‘.
`y`	Integer column indices or name of dependent variable
`cov`	Integer column indices or name of covariate variables
`factors`	Integer column indices or names of variables to be treated as factor
`model`	regression model, see `lm`, `glm`, `coxph` for more details
`time`	Integer column indices or name of survival time, used in cox regression, see `coxph` for more details
`confint_glm`	A character, ’default’ or ’profile’. The default method for ’glm’ class to compute confidence intervals assumes asymptotic normality `confint`, you can also use profile likelihood method `confint.glm`, but it is pretty slow. In this case you could specify ’default’ for speed.
`cov_show`	A logical, whether to create covariates result, default FALSE

Value

The return result is a concentrated result in a data.frame.

Index