Package 'ormPlot'

May 9, 2026

Type Package

Title Advanced Plotting of Ordinal Regression Models

Version 0.3.6

Maintainer Richard Meitern <richard.meitern@ut.ee>

Description An extension to the Regression Modeling Strategies package that facilitates plotting ordinal regression model predictions together with confidence intervals for each dependent variable level.
It also adds a functionality to plot the model summary as a modifiable object.

License MIT + file LICENSE

URL https://doi.org/10.1186/s12889-019-8072-7

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports ggplot2 (>= 3.1.0), rms (>= 5.1.3), gtable (>= 0.3.0), grid (>= 3.5.0)

Suggests testthat (>= 2.1.0), vdiffr (>= 0.3.0), knitr (>= 1.22), rmarkdown (>= 1.13), pander (>= 0.6.3)

RoxygenNote 7.2.1

VignetteBuilder knitr

NeedsCompilation no

Author Richard Meitern [aut, cre] (ORCID: <https://orcid.org/0000-0002-2600-3002>)

Repository CRAN

Date/Publication 2023-09-13 14:40:02 UTC

Contents

`convert_arg` *Function to convert any input to string vector*

Description

Function to convert any input to string vector

Usage

```
convert_arg(x)
```

Arguments

x	string, object name or vector of these

Value

vector of strings

`educ_data` *Morfometrics of children*

Description

A dataset containing the standardized residuals of Estonian schoolchildren.

Usage

```
educ_data
```

Format

A data frame with 11032 rows and 10 variables:

educ_3 highest obtained educational level

Rural location of school (rural or urban)

sex gender of the child)

max_SEP_3 highest parental profession

n_siblings number of children in the family

cran_rzs cranial volume (residuals to age an birth date)

height_rzs height (residuals to age an birth date)

FW_rzs face width (residuals to age an birth date)

YOB year of birth

YOBc centered but not scaled year of birth (YOB)

`join_ggplots` *Join two ggplot objects side by side*

Description

Function to get aligned table of two ggplot objects

Usage

```
join_ggplots(
  leftplot,
  rightplot,
  plot.widths = c(0.5, 0.5),
  title = "Odds Ratio"
)
```

Arguments

`leftplot`	the left side plot
`rightplot`	the plot on the right
`plot.widths`	the relative widths of the left and right plot should be a vector (`c()`) with 2 elements that sum to 1 defaults to equal widths
`title`	the tile row of the drawn plot

Examples

```
set.seed(123)
#load the libraries
library(rms)
library(ormPlot)
library(ggplot2)

#make the datadist
dd<-rms::datadist(educ_data)
options(datadist="dd")

#create the model
cran_model <- orm(educ_3 ~ YOBc + sex + height_rzs + n_siblings  + cran_rzs, data = educ_data)

#the antilog true produces odd ratios (default value for orm and lrm)
s<-summary(cran_model, antilog = TRUE)

#set the plotting default theme (optional)
theme_set(theme_classic())

#return modifiable ggplots
plots<-forestplot(s, return_ggplots = TRUE )

#modify like any ggplot2 object
table<-plots[[1]] +  theme(axis.text=element_text(size = 12),
                           axis.line.x = element_line(color = "red", size = 1),
                           axis.text.y = element_blank())

graph<-plots[[2]] + theme(axis.line = element_line(color = "red", size = 1),
                          axis.text.y = element_text())

#join the graphs
join_ggplots(  graph, table, title = "", plot.widths = c(0.6,0.4))
```

`oddstable` *Get row names from odd an values form even columns*

Description

Get row names from odd an values form even columns

Usage

```
oddstable(x)
```

Arguments

`x`	a `matrix` with even number of rows

oddstable_graph *Make a ggplot table*

Description

Function to get a ggplot table from a matrix

Usage

```
oddstable_graph(
  x,
  digits = 3,
  theme = ggplot2::theme_get(),
  header = NULL,
  row.names.y = NULL
)
```

Arguments

x	a matrix or a data.frame
digits	the number of significant digits to display
theme	the desired ggplot2 theme
header	names of the table columns
row.names.y	new names for the variable rows

ormPlot *ormPlot: Plotting ordinal regression models from* rms

Description

The package is an extension to the rms package that facilitates plotting the ordinal regression orm model objects. The aim is to get ggplot2 plots that are modifiable

Details

The ormPlot package provides two categories of important functions: forestplotting the summary and plotting the predictions

Summary plotting

The forestplot function facilitates plotting the summary.rms objects resulting from the orm or lrm model

See exported methods for more details:

- plot.summary.rms
- forestplot
- join_ggplots

Prediction plotting

The prediction plotting function facilitates plotting the orm objects using the results got from Predict function. In particular it adds confidence intervals to orm prediction plots.

See exported methods for more details:

- plot.orm
- predict_with_ci

Data

educ_data data about morfometrics of schoolchildren born between 1937-1962 in Estonian territory. see also the citation("ormPlot") article

---

orm_graph *Make a ggplot figure*

Description

Function to get a ggplot figure from a matrix x

Usage

```
orm_graph(
  x,
  theme = ggplot2::theme_get(),
  header = NULL,
  row.names.y = NULL,
  shape = 19,
  limits = NULL,
  breaks = c(0.5, 1, 1.5, 2, 3, 4)
)
```

Arguments

x	a matrix or a data.frame
theme	the desired ggplot2 theme
header	names of the table columns
row.names.y	new names for the variable rows
shape	point shape, see aes_linetype_size_shape
limits	the x axis limits as a vector, see also: scale_continuous
breaks	the x axis breaks as a vector,help see also: scale_continuous

plot.lrm *Plot the prediction with confidence intervals*

Description

This function plots the model predictions given that all variables that are not included in the plot are kept constant. Hence it requires at least one variable to produce a plot. returns a ggplot object that can be further customized like any other ggplot

Usage

```
## S3 method for class 'lrm'
plot(x, ...)
```

Arguments

x	an object created by Predict
...	additional parameters that will be passed to Predict

Value

a ggplot plot object

See Also

Predict, datadist, orm

Examples

```
#load the libraries
library(rms)
library(ormPlot)

#make the datadist
dd<-datadist(educ_data)
options(datadist='dd')

#create the model
cran_model <- orm(educ_3 ~ Rural + sex + max_SEP_3 + cran_rzs, data = educ_data)

#plot the predictions of the model for varying one variable only
plot(cran_model, cran_rzs)

#customize the plotting varying all variables
plot(cran_model, cran_rzs,
      plot_cols = max_SEP_3,
      plot_rows = c(Rural, sex),

      #setting new x-label (optional)
```

```
        xlab = "Cranial volume (residuals to age an birth date)",

        #setting new facet labels (optional)
        facet_labels = list(Rural = c("Urban", "Rural"),
                            sex = c("Boys","Girls"))
        )
```

---

plot.orm *Plot the prediction with confidence intervals*

Description

This function plots the model predictions given that all variables that are not included in the plot are kept constant. Hence it requires at least one variable to produce a plot. returns a ggplot object that can be further customized like any other ggplot

Usage

```
## S3 method for class 'orm'
plot(
  x,
  xval,
  plot_cols = c(),
  plot_rows = c(),
  label_with_colname = TRUE,
  facet_labels = NULL,
  xlab = NULL,
  ylab = NULL,
  np = 100,
  fun = stats::plogis,
  boot.type = "bca",
  conf.int = 0.95,
  ...
)
```

Arguments

x	an object created by Predict
xval	The model value plotted on the x axis
plot_cols	A vector of strings with other model components that should be plotted. These are put on columns.
plot_rows	A vector of strings with other model components that should be plotted. These are put on rows.
label_with_colname	Should he variable name also be included on plot row and column names
facet_labels	A named list of new names for variables on rows and columns

xlab	A custom x-axis value (if specified)
ylab	A custom y-axis value (if specified)
np	the number of equally-spaced points computed for continuous predictors that vary, i.e., when the specified value is . or NA
fun	an optional transformation of the linear predictor. Specify fun='mean' if the fit is a proportional odds model fit and you ran bootcov with coef.reps=TRUE. This will let the mean function be re-estimated for each bootstrap rep to properly account for all sources of uncertainty in estimating the mean response. fun can be a general function and can compute confidence limits (stored as a list in the limits attribute) of the transformed parameters such as means.
boot.type	set to 'bca' to compute BCa confidence limits or 'basic' to use the basic bootstrap. The default is to compute percentile intervals
conf.int	confidence level (highest posterior density interval probability for Bayesian models). Default is 0.95. Specify FALSE to suppress.
...	additional parameters that will be passed to Predict

Value

a ggplot plot object

See Also

Predict, datadist, orm

Examples

```
#load the libraries
library(rms)
library(ormPlot)

#make the datadist
dd<-datadist(educ_data)
options(datadist='dd')

#create the model
cran_model <- orm(educ_3 ~ Rural + sex + max_SEP_3 + cran_rzs, data = educ_data)

#plot the predictions of the model for varying one variable only
plot(cran_model, cran_rzs)

#customize the plotting varying all variables
plot(cran_model, cran_rzs,
      plot_cols = max_SEP_3,
      plot_rows = c(Rural, sex),

      #setting new x-label (optional)
     xlab = "Cranial volume (residuals to age an birth date)",

     #setting new facet labels (optional)
```

```
    facet_labels = list(Rural = c("Urban", "Rural"),
                        sex = c("Boys","Girls"))
    )
```

`plot.summary.rms` *Forest Plot of an rms model summary*

Description

Convenience function to create a plot of the `orm` model summary. For further customizing the plots use `return_ggplots = TRUE` This will create 2 `ggplot2` objects that can be joined with the `join_ggplots` commands.

Usage

```
## S3 method for class 'summary.rms'
plot(x, ...)

forestplot(
  x,
  return_ggplots = FALSE,
  plot.widths = c(0.5, 0.5),
  title = "Odds ratio",
  digits = 3,
  shape = 19,
  header = NULL,
  limits = NULL,
  breaks = c(0.5, 1, 1.5, 2, 3, 4),
  theme = ggplot2::theme_get(),
  row.names.y = NULL
)

forestplot.default(x, ...)

forestplot.summary.rms(x, ...)
```

Arguments

`x`	result of a `summary` command on `orm` or `lrm` model ie a `summary.rms` class object
`...`	see parameters of method `forestplot`
`return_ggplots`	if `TRUE` the function returns 2 ggplot objects in a list instead of drawing a table-grid
`plot.widths`	the relative widths of the left and right plot should be a vector (`c()`) with 2 elements that sum to 1 defaults to equal widths
`title`	the tile row of the drawn plot

digits	the number of significant digits to display
shape	point shape, see aes_linetype_size_shape
header	names of the table columns
limits	the x axis limits as a vector, see also: scale_continuous
breaks	the x axis breaks as a vector,help see also: scale_continuous
theme	the desired ggplot2 theme
row.names.y	new names for the variable rows

Examples

```
set.seed(123)
#load the libraries
library(rms)
library(ormPlot)
library(ggplot2)

#make the datadist
dd<-rms::datadist(educ_data)
options(datadist="dd")

#create the model
cran_model <- orm(educ_3 ~ YOBc + sex + height_rzs + n_siblings  + cran_rzs, data = educ_data)

#the antilog true produces odd ratios (default value for orm and lrm)
s<-summary(cran_model, antilog = TRUE)

#set the plotting default theme (optional)
theme_set(theme_classic())

#show simply the result
forestplot(s)

#return modifiable ggplots
forestplot(s, return_ggplots = TRUE )

#new row names and header
newnames <- c("Year of birth", "Height", "Number of children", "Cranial volume", "Sex" )
newhead <- c("Odds Ratio", "CI 5%", "CI 95%" )

#adjust also the relative plot widths and change the color and shape
newtheme <- theme_classic() + theme(text = element_text(color = "red", size = 12),
                                    line = element_line(color= "red"),
                                    rect = element_rect(color="red"))

forestplot(s, row.names.y = newnames, header = newhead,
           plot.widths = c(0.6,0.4), shape = 17,
           theme = newtheme)
```

predict_with_ci *Create a Prediction data.frame with confidence intervals*

Description

returns a data.frame object similar to the Predict however it adds a column dependent that lists all factor levels with appropriate confidence intervals calculated for each level. It is similar to predict.lrm with type="fitted.ind" but also generates selected confidence intervals.

Usage

```
predict_with_ci(
  x,
  ...,
  np = 100,
  fun = stats::plogis,
  conf.int = 0.95,
  boot.type = "bca"
)
```

Arguments

x	an object created by Predict
...	One or more variables to vary, or single-valued adjustment values. Specify a variable name without an equal sign to use the default display range, or any range you choose (e.g. seq(0,100,by=2),c(2,3,7,14)). The default list of values for which predictions are made is taken as the list of unique values of the variable if they number fewer than 11. For variables with $>$ 10 unique values, np equally spaced values in the range are used for plotting if the range is not specified. Variables not specified are set to the default adjustment value limits[2], i.e. the median for continuous variables and a reference category for non-continuous ones. Later variables define adjustment settings. For categorical variables, specify the class labels in quotes when specifying variable values. If the levels of a categorical variable are numeric, you may omit the quotes. For variables not described using datadist, you must specify explicit ranges and adjustment settings for predictors that were in the model. If no variables are specified in ..., predictions will be made by separately varying all predictors in the model over their default range, holding the other predictors at their adjustment values. This has the same effect as specifying name as a vector containing all the predictors. For rbind, ... represents a series of results from Predict. If you name the results, these names will be taken as the values of the new .set. variable added to the concatenated data frames. See an example below.
np	the number of equally-spaced points computed for continuous predictors that vary, i.e., when the specified value is . or NA
fun	an optional transformation of the linear predictor. Specify fun='mean' if the fit is a proportional odds model fit and you ran bootcov with coef.reps=TRUE.

This will let the mean function be re-estimated for each bootstrap rep to properly account for all sources of uncertainty in estimating the mean response. `fun` can be a general function and can compute confidence limits (stored as a list in the `limits` attribute) of the transformed parameters such as means.

`conf.int` confidence level (highest posterior density interval probability for Bayesian models). Default is 0.95. Specify `FALSE` to suppress.

`boot.type` set to `'bca'` to compute BCa confidence limits or `'basic'` to use the basic bootstrap. The default is to compute percentile intervals

Value

a `data.frame`

See Also

`Predict`,`orm`, `predict.lrm`

Examples

```
set.seed(123)
#load the libraries
library(rms)
library(ormPlot)

#make the datadist
dd<-rms::datadist(educ_data)
options(datadist="dd")

#create the model
cran_model <- orm(educ_3 ~ Rural + sex + max_SEP_3 + cran_rzs, data = educ_data)

#get the predictions of the orm model with confidence intervals for all levels
predictiondf<-predict_with_ci(cran_model, cran_rzs, Rural, sex, max_SEP_3)
#show the predictions head
head(predictiondf)

#get the predictions of the orm model with confidence intervals for sex only
predictiondf_sex<-predict_with_ci(cran_model,  sex)
#show the predictions head
head(predictiondf_sex)
```

Index